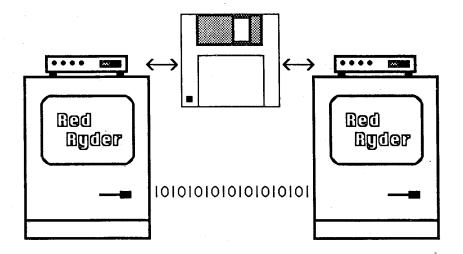
Red Ryder.

An Asynchronous Serial Communications Program For The Apple Macintosh Using A Modem Or Direct Connection.



Version 10.0 By Scott Watson

Red Ryder program and documentation is Copyright ©, 1984, 1985,1986,1987 by The FreeSoft Company, 150 Hickory Drive, Beaver Falls, PA 15010. All rights reserved.

The name Red Ryder is a registered trademark of Red Ryder Enterprises, Inc., and is used with permission.

IMPORTANT INFORMATION! YOU MUST READ AND AGREE TO THE ENCLOSED SINGLE-COMPUTER END USER SOFTWARE LICENSE AGREEMENT BEFORE ATTEMPTING TO USE THE ENCLOSED MASTER DISK. BY USING THE ENCLOSED MASTER DISK, YOU ARE AGREEING TO THE TERMS AND CONDITIONS OF THAT DOCUMENT.

Dear Customer:

Thank you for your order of Red Ryder 10.3. Please note that the enclosed Master disk is NOT a bootable system disk. You must start up your Macintosh with a disk containing a System Folder, and then insert the enclosed Master disk. Don't do work with the Master, make a backup of it and put the Master disk in a safe place in case your backup craps out. Your Master disk will entitle you to obtain future updates.

General Electric has graciously extended a free subscription to their GEnie network for all of our customers. This means they waive their normal \$29.95 signup fee. This is available to all customers in the U.S. and Canada. Other customers will receive mail update notices, but U.S. and Canada customers should immediately do the following so they can enter the FreeSoft RoundTable on GEnie, which is your sole source (except by telephone) for future update information.

- 1) Using 300 or 1200 baud, NO parity, 8 databits, 1 stopbit, and HALF duplex (not FULL!), call the following toll free number: 1 (800) 638-8369. In Toronto, Canada call 1 (416) 858-1230; Calgary, Alberta, Canada call 1 (403) 232-6121; Montreal, Canada call 1 (514) 333-1117; Vancouver, B.C. Canada call 1 (604) 437-7313.
- 2) When connected, immediately type 3 'H' characters and press the Return key.
- 3) After a couple of seconds you'll get a prompt of "U#=". You should type: XJM11979, FREESOFT and press the Return key.
- 4) You'll now be taken on a tour of GEnie and asked to fill in your billing information. The "800" number does not give you access to the actual GEnie network, but is for subscription purposes only. You can and should check out what your local GEnie access number (s) are while doing this procedure. Those numbers will be what you'll call after receiving your account to log into GEnie.
- 5) After supplying your billing information, you'll be contacted voice by GEnie within the next week with your new account number and password. After you get this, you can log onto GEnie through your local access number. Once on, you should type the word "FREESOFT" at any GEnie menu prompt.
- 6) You'll be teleported to the FreeSoft RoundTable, but you won't be allowed to get in. Don't worry, you have to be turned away at least once before I can let you in. Type "MAIL" at the next menu to get the GEnie mail section.
- 7) Send a short note to me (address SWATSON) saying simply "Let me in". To send a letter you must type an asterisk character ('*') and a 's' character at the beginning of a new line like this: *s.
- 8) We usually do the unlocking each business day, but don't get upset if it takes us up to three business days. You won't receive any notice about being unlocked, but can check quickly by typing "FREESOFT" at any GEnie menu to attempt to enter the FreeSoft RoundTable.

SDECLE-CONFUTER AND USER SOFTWARE LICENSE ACCUMENT

The FreeSoft Company ("FreeSoft") makes all PresEoft application, demonstration, system software and technical information (the "Software") available subject to the following license. You are responsible for selecting the appropriate Software (computer programs and related discussion) to meet your needs, and for its installation, use and results.

This license allows you to:

- 1. Use the Software only on a single computer.
- 2. Make one copy of the Software in any machine-readable form for backup purposes. The Software is protected by United States copyright law. You must not remove or modify the FreeSoft copyright notice on any copy of the Software.
- Configure the Software for your own use by adding or removing fonts, desk accessories and/or device drivers.

THE SOFTWARE MAY BE DISTRIBUTED ONLY BY AUTHORIZED DISTRIBUTORS. IT MAY BE USED ONLY FOR PERSONAL OR NON-COMMERCIAL USES AND MAY NOT BE REDISTRIBUTED OR USED FOR COMMERCIAL PURPOSES WITHOUT AN EXPRESS SOFTWARE DISTRIBUTION LICENSE FROM FREESOFT. THESE LICENSES ARE AVAILABLE FROM FREESOFT.

YOU MAY NOT MODIFY, NETWORK, RENT, LEASE, LOAN, SELL OR DISTRIBUTE THE SOFTWARE, OR ANY COPY, IN WHOLE OR IN PART. YOU UNDERSTAND THAT UNAUTHORIZED REPRODUCTION OF A COPY OF THE SOFTWARE OR UNAUTHORIZED TRANSFER OF ANY COPY OF THE SOFTWARE MAY SUBJECT YOU TO A LAWSUIT FOR DAMAGES, INJUNCTIVE RELIEF, AND ATTORNEY'S FEES.

FreeSoft reserves all rights not expressly granted to you.

EXPORT LAW ASSURANCES

You agree and certify that neither the Software nor any other technical data received from FreeSoft or the direct product thereof, is being or will be downloaded, shipped, transferred or reexported, directly or indirectly to Country Groups Q,S,W,Y, or Z (see Attachment 1) or Afghanistan or the People's Republic of China, without prior approval from FreeSoft and either a validated export license or written permission from the U.S. Office of Export Administration.

TERM

The license is effective until terminated. You may terminate it at any time by destroying the Software together with all copies. The license will also terminate if you fail to comply with any of the terms or conditions of this Agreement. You agree upon such termination to destroy all copies of the Software.

DISCLAIMER OF WARRANTY

THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO ITS MERCHANTABILITY OR ITS FITNESS

FOR ANY PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU (AND NOT FREESOFT OR AN FREESOFT AUTHORIZED REPRESENTATIVE) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

FreeSoft does not warrant that the functions contained in the Software will meet your requirements or that the operation of the Software will be uninterrupted or error-free or that defects in the Software will be corrected.

SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

LIMITATION OF REMEDIES

IN NO EVENT WILL FREESOFT BE LIABLE TO YOU FOR ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE ANY SOFTWARE EVEN IF FREESOFT OR AN AUTHORIZED FREESOFT REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

FreeSoft's liability to you for actual damages for any cause whatsoever, and regardless of the form of the action, will be limited to the greater of \$50 or the money paid for the Software that caused the damages or that is the subject matter of, or is directly related to, the cause of action.

GOVERNMENT END USERS

If this Software is acquired by or on behalf of a unit or agency of the United States Government this provision applies. This Software: (a) was developed at private expense, and no part of it was developed with government funds, (b) is a trade secret of FreeSoft for all purposes of the Freedom of Information Act, (c) is "commercial computer software" subject to limited utilization as provided in the contract between the vendor and the governmental entity and (d) in all respects is proprietary data belonging solely to the FreeSoft Company.

For units of the Department of Defense (DoD), this Software is sold only with "Restricted Rights" as that term is defined in the DoD Supplement to the Federal Acquisition Regulations, 52.227-7013 (b) (3) (ii) and:

Use, duplication or disclosure is subject to restrictions as set forth in subdivision (b) (3) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. Manufacturer: The FreeSoft Company, 150 Hickory Drive, Beaver Falls, PA 15010

If this Software was acquired under a GSA Schedule the Government has agreed to refrain from changing or removing any insignia or lettering from the Software that is provided or from producing copies of manuals or disks (except one copy for backup purposes) and:

(1) Title to and ownership of this Software and documentation and any reproductions thereof shall remain with The FreeSoft Company.

- (2) Use of this Software and documentation shall be limited to the facility for which it is acquired; and
- (3) If use of the Software is discontinued to the installation specified in the purchase/delivery order and the Government desires to use it at another location, it may do so by giving prior written notice to The FreeSoft Company, specifying the type of computer and new location site.

Government personnel using this FreeSoft software, other than under a DoD contract or GSA Schedule, are hereby on notice that use of this Software is subject to restrictions which are the same as or similar to those specified above.

CONFIDENTIALITY

You agree not to directly or indirectly divulge to any person or entity the Software, any trade secrets related to the Software, or any information relating to FreeSoft's product plans, product designs, product costs, product prices, marketing plans, business opportunities, personnel research, development or know-how, until such time as FreeSoft makes such information available to the public without restriction.

GENERAL

Any attempt to network, rent, lease, or sublicense the Software or to transfer any of the rights, duties or obligations under this Agreement is void.

This Agreement will be construed under the laws of the state of Pennsylvania, except for that body of laws dealing with conflict of laws. If any provision of this Agreement shall be held by a court of competent jurisdiction to be contrary to law, that provision will be enforced to the maximum extent permissible, and the remaining provisions of this Agreement shall remain in full force and effect.

ATTACHMENT 1

(Note that this list is current as of January 28, 1987. You should check with the Office of Export Administration for any subsequent updates of revisions)

Country Group Q Libya Country Group S Romania Country Group W Hungary Poland

Country Group Y
Albania
Bulgaria
Czechoslovakia
Estonia
German Democratic Republic
Laos
Latvia
Lithuania
Mongolian People's Republic
USSR

Country Group Z Cuba Kampuchea North Korea Vietnam

Table Of Contents

Prologue	1
What's In A Filename?	4
The Terminal Display Window	
Serial Port Settings	12
The General Status Bar	16
Keyboard Specifics	18
The Buffered Keyboard Status Bar	22
The Macros Status Bar	24
The VT100 Status Bar	28
Redefining The Standard VT100 Special Keys	29
Customizing The Terminal Emulation	32
Terminal Emulation Specifics	34
Using A Printer With Red Ryder	38
Connecting To Remote Systems	40
Phonebooks	42
Getting It From There To Here Or From Here To There	46
Sending Text Files	47
Receiving Text Files	51
Sending And Receiving Files Using Error Correcting Protocols	53
Getting Rid Of Files	62
Getting Personal	63
Settings Files	65
Host Mode	66
Using International Or Special Character Sets	70
Procedure Files	
Procedure Commands	83
Procedure File Examples	124
Letting Red Ryder Write A Procedure File For You	131
The GETPARAM And PUTPARAM Prodedure Commands	134
Index Of Menu Commands	144
Index Of Procedure Commands	146

Proloque

There's an old joke about a fellow who gets lost and stops his car to ask a farmer how to get to the City. The farmer thinks a moment, spits, and then replies "Can't get there from here."

For some reason, that rancid old line keeps coming back to me every time I'm asked about just how Red Ryder came to be. Let me explain.

Red Ryder is a fluke. When I bought my 128K Macintosh in early 1984, I had a gut feeling that it was the riskiest, and probably the poorest investment I'd ever made. The total inventory of available software was about a half dozen products, including MacPaint, and MacWrite (which at that time came free with the Macintosh). The only development language available was Microsoft BASIC 1.0. I'd shelled out nearly two grand for something that did almost nothing.

Being a modem fanatic, the first thing I wanted to do was write a program that would allow me to download MS-BASIC programs from IBM-PC bulletin board systems so that I could convert them to run on a Mac, and in doing so, increase my library of Mac software. Red Ryder 1.0 was born. It was written with MS-BASIC 1.0.

Despite its severe limitations, it quickly became clear that Red Ryder 1.0 was useful to others besides myself. Having had an extraordinarly bad experience with a software publisher in the past, I was more than a little hesitant to take that path again. About that time, the concept of user supported software was becoming popular because of an IBM-PC communications program called PC-TALK. PC-TALK was wildly successful, and if nothing else, proved to me that I wasn't the only one who had thousands of dollars worth of lousy software on my shelves. Clearly, the computer industry was hungry for decent software at a decent price. I felt that much of the software at that time was being sold as if it were underarm deordorant or toothpaste, complete with flashy packaging, deceptive advertising, and distributors interested more in commissions than integrity.

Red Ryder became a passionate experiment. While I knew I couldn't defeat Samson, I took on a "what the hell" attitude that convinced me that it would at least be fun to give him the finger. My theory was that it was time for a software product that had thousands of project managers - the consumers. An ultimate test of capitalism. Break some rules, dammit! I became a technocrat who despised the army of corporate bean counters who were bastardizing our personal revolution.

I quit my job and decided to start up The FreeSoft Company. My philosopy was (and still is) simple: "If my product isn't useful, or if I charge too much for it, I deserve to go out of business." Don't get me wrong, I wasn't naive. I knew

about and practiced what some companies refer to as "solution princy".

My rationale for using software before I paid for it was that I became entirely convinced that the advertising and P.R. flacks had fine tuned the art of deception into a science. Unlike an automobile, I knew that if I test drove a piece of software and was absolutely convinced that it performed the job I needed, I couldn't later be stuck with a lemon. I became an outlaw out of financial necessity. For moral reasons, then, I had no other choice than to market Red Ryder as shareware.

This wasn't like walking through Times Square with your wallet on your head. I never felt like I had to protect myself against common thugs by doing this. I knew that I would be dealing with a higher class of people. People who understand right from wrong. Red Ryder was like a modernized version of Abby Hoffman's "Steal This Book", except that it was marketed by a conservative capitalist from Missouri.

If it's good, and you use it, you are morally and ethically bound to pay for it. I like that spirit. I embrace it. I depend upon it. Most of us do.

The shareware marketing concept is simple. It is too prove to you that I am not interested in cheating you when we do business. I am convinced that the first nine versions of Red Ryder have proved that. I am now confident that the Macintosh owners are also confident of that. There are many good reasons why this version could not be marketed as shareware (the size of this manual alone is #1 on the list). But, before I get accused of selling out, I wish to submit that I have paid my dues.

But like I said, I'm not naive. Please continue to give out copies of version 9.4 to anyone and everyone as the "trial version". If you give out this version without the manual, a person will no doubt become quickly lost, and will never know what Red Ryder can do for them. And... it makes technical support Hell - believe me. Please understand that I can better serve you by working on the next version of Red Ryder, as well as other projects, than I can explaining to someone for the 10,000th time what would be clear if they just had this documentation. Not only will you be hurting a person by giving them a copy of this version without the documentation, you will hurt yourself in the long run because my time will have to be spent on something else than new product development. In other words, by protecting me, you protect your investment.

As I write this paragraph, I know that this milestone is very close to being completed. More hours of development have gone into this version than all of the others combined, in which time I have worked myself into the hospital twice, have given up the "luxuries" of a proper diet and something called sleep, and have pulled this version out of beta test and back into development six times because I wasn't <u>completely</u> happy with it. It's been a long damn road. You can't get there from here, as the saying goes.

One thing that won't change about Red Ryder is the fact that I will never consider it "finished", and I rely heavily on you to tell me what is right and wrong about it. I beg you to write and tell me what you want to see in future versions. I read all of your suggestions and take them to heart. I only wish that I could be there to see the smiles on those peoples' faces who will see their wish list (or at least part of it) implemented in this version. This version is definitely a milestone, but it in no way marks the end of the trail.

I wish you the best results from your pursuits, and I hope to run into you soon online or in person.

Sincerely, Scott Watson

What's In A Filename?

In various places throughout this manual, the word **filename** is used to **indicate** a group of characters that describe the location of a file (either a **Macintosh** application or document) on a disk. There are different ways of doing this, depending on the kind of filing system you are using.

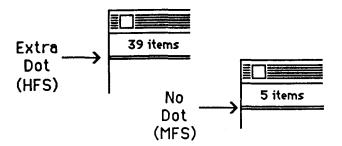
The original Macintosh filing system was called MFS (Macintosh Filing System), which worked well for floppy disks. Folders that are displayed in a Finder window are for cosmetic purposes only, and they do not play a part in constructing a filename for Red Ryder. An MFS filename consists of the disk volume name (which appears under the disk icon in the Finder), a colon character, and the name of the file (which appears under the file's icon in the Finder).

Under MFS, a file with the name of "Junk" that resides on a disk named "Stuff" would be given the filename "Stuff:Junk"

When hard disks became common in the Macintosh world, it became necessary for Apple to devise a more efficient filing system, which they named HFS (Hierarchical Filing System). Under HFS, the names of folders which lie between the desktop (the top level window in the Finder) and the file are significant and must be included in the filename in the order they appear and seperated by colon characters. An example should clarify this.

Under HFS, a file named "Junk" is inside of a folder named "Garbage". The "Garbage" folder is inside of another folder named "Trash". The "Trash" folder lies on the desktop of a disk named "Stuff". The proper filename would then be: "Stuff:Trash Garbage: lunk"

How do you know if a disk uses HFS or MFS? Open the disk icon in the Finder and look at the upper left hand corner of the window. There is a small division between the two lines that appear just below the phrase that tells you the number of items on the disk. A HFS disk will have an extra dot at the far left of this division, and a MFS disk will have no dots in this division. The illustration below shows an example of both kinds of disks.



It's important to know what kind of disks you are working with so that you don't construct a faulty filename and get the dreaded "File Not Found" error message.

One last tip. If you are sure that you've spelled everything correctly and are positive a filename should work but doesn't, you should be aware that the Finder allows you to (inadvertently or otherwise) put leading or trailing spaces in disk, folder, and file names. If these are present, they <u>must</u> be included in the filename you present to Red Ryder. So, if problems persist, try renaming the entities in a filename to make sure none of these "ghost" spaces exist.

The Terminal Display Window

When Red Ryder is operating, two major components are immediately visible in your display screen, the menu bar at the top of the screen, and the Terminal Display Window immediately below that.

The Terminal Display Window is where most of the action in Red Ryder takes place. It consists of several parts, each of which will be discussed in detail. The entire Terminal Display Window can be made invisible by selecting the Hide Terminal Window command under the Local menu. When it's hidden, you can bring back the Terminal Display Window by choosing Hide Terminal Window again. You probably will not make much use of this command, but be aware that Procedure files may hide the Terminal Display Window at certain times to keep the display uncluttered. Therefore, if the window disappears when you don't want it to, you can choose Hide Terminal Window (to uncheckmark it) to bring it back.

By choosing the **Bring Other Open Windows To Top** command from under the **Local** menu, you can bring any other open windows (such as those used by desk accessories) to the top of the desktop and send the Terminal Display Window to the rear.

- The Title Bar
- The Close Box
- The Zoom Box
- The Size Box
- The Horizontal Scroll Bar
- The Vertical Scroll Bar
- The Text Display Area
- The Status Bar

The Title Bar

In the center of the Title Bar is displayed the version number of the Red Ryder application you are using. When a compiled Procedure File is executing (these are discussed later in this manual), the file name of the Procedure File is displayed there. The Close Box and Zoom Box are also contained in the Title Bar. The Terminal Display Window can be dragged around the Macintosh video screen in a standard manner by clicking and dragging in the Title Bar.

The Close Box

The lefthand corner of the Title Bar contains the Close Box. Clicking your mouse and releasing in the Close Box is equivalent to choosing **Quit** from under Red Ryder's **File** menu. Execution is terminated and you are returned to the Finder.

The Zoom Box

On all Macintosh models except the 128K and 512K (unenhanced), a Zoom Box

is displayed in the righthand corner of the Title Bar. Clicking in the Zoom Box will alternate between growing the window to the full screen size, and shrinking it to the last size you were using.

The Size Box

The Size Box is located in the bottom righthand corner of the Terminal Display Window. It is used in a standard manner to enlarge or shrink the size of the Terminal Display Window. In case you're wondering (or have an awfully large display screen), the largest Terminal Display Screen you may have is about 32,000 pixels wide by 32,000 pixels high. This should be sufficient for most users for the near future.

The Horizontal Scroll Bar

The Horizontal Scroll Bar is located at the bottom of the Terminal Display Window. It is active when the maximum column width number of the widest character is too wide to fit entirely in the Terminal Display Window. When this occurs, the Horizontal Scroll Bar is used to scroll left and right as needed to see all of a line.

Here's a trick. Rarely if ever will you receive a full line of the widest character in a font, so there's quite a bit of unused "white" space at the rightmost part of the scrolling region when you use a proportionally sized font. If you hold down the Option key and click your mouse in the "Page Right" area of the Horizontal Scroll Bar (the grey area between the thumb and right arrow box), Red Ryder will scroll right to the point where the widest character in the displayed lines will be flush with the right edge of the Terminal Display Window. Holding down the Option key and clicking in the "Page Left" part of the Horizontal Scroll Bar is equivalent to dragging the thumb all the way to the left.

The Vertical Scroll Bar

The Vertical Scroll Bar is located on the righthand edge of the Terminal Display Window. It is used to scroll back through lines of text already received, but no longer displayed in the Terminal Display Window. Red Ryder defaults to remembering the last 24 lines of text received, but you can extend this to cause Red Ryder to remember as many screens of received text as you have free memory to hold. To change the number of screens Red Ryder will remember in this saved screens buffer, select Set Saved Screens Buffer Size... from under the Customize menu. On a 512K Macintosh, you could comfortably set this buffer size to hold 25 to 50 screens if all free memory is available to Red Ryder.

I recommend that you not set this buffer to an extraordinarily large size without purpose. Otherwise, functions that erase all screens (which include the **Clear All Screens** choice under the **Local** menu, or changing the font, emulation, or column width) can take an annoying amount of time.

Here's another trick. If you save 20 screens of text in the saved screens buffer, there can be a lot of empty "white space" in the buffer until 20 screens have

been received. If you hold down the Option key and click in the "Page Up" portion of the Vertical Scroll Bar (the grey area between the thumb and the up arrow box), Red Ryder will position the thumb to the first non-empty line in the buffer (the first line of text received). Holding down the Option key and clicking in the "Page Down" area of the Vertical Scroll Bar is equivalent to dragging the thumb all of the way to the bottom. Typing any character on the keyboard when the Vertical Scroll Bar is anywhere but at its bottom position will automatically move the Vertical Scroll Bar to its bottom position.

The Text Display Area

The Text Display Area is located below the Status Bar and above the Horizontal Scroll Bar. This area is where all data received through the serial port (except during XMODEM, Kermit, or CompuServe 'B' protocol file transfers) is displayed.

You can quickly locate a desired string of characters by choosing **Find Text...** from under the **Edit** menu. Red Ryder will search from the top of the Saved Screens Buffer to the bottom and point out the first occurance (if there is one) of the string of characters you specify. Whenever the text display area is scrolled to display the found text, a zooming rectangle is displayed to help you locate the found text. Whether or not the window scrolls, the found text will flash several times. You can continue searching through the buffer by choosing **Find Same Text Again** from under the **Edit** menu repeatedly until the search reaches the bottommost screen of text.

You can erase just the text in the Terminal Display Area by choosing Clear Display Screen from under the Local menu. To erase all text in the Saved Screens Buffer (and Terminal Display Area), choose Clear All Screens from under the Local menu.

The text in this area is not editable as in a word processor, but you can copy text from the screen to the Clipboard, or paste text from the clipboard to the modem. Text is selected by dragging the mouse across it as with a word processor. As text is selected, it is shown inverted (white characters on a black background). You can scroll horizontally or vertically if the appropriate scroll bar is active by dragging your mouse outside of the Text Display Area in the direction you wish to scroll.

A quick way to select a block of text is to click your mouse once at the beginning of the block, then hold down the Shift key and click once at the end of the block. The entire block will then be shown as selected.

If you select text and then change your mind, you can click your mouse anywhere in the terminal display window and the selection will be undone. Once text is selected, you may choose either Copy, Copy Table, Print Selected Text, or Append To Archive File from under the Edit menu.

The Copy And Copy Table Commands
Under the Edit menu, the Copy command will copy all of the selected text into

the Clipboard file. The Copy Table command works in much the same way, except that it helps you later import data from the screen into a spreadsheet or database application. Copy Table looks for more than one space character in a row, and converts all of the consecutive spaces to a single tab character. The data in the Clipboard is then referred to as "tab delimited", which many spreadsheets and database programs allow to be imported (consult the user manuals or manufacturers of these programs - not FreeSoft - for more details on how to import tab delimited data). Copy Table leaves all single spaces alone.

The Print Selected Text Command

Under the Edit menu, the Print Selected Text command will send the selected data to your printer. See the "Using A Printer With Red Ryder" chapter in this manual for more details about printing. If the selected text is more than 100 characters long, Red Ryder will save the selected data in a temporary text file, and then automatically use the Print TEXT File... command (explained later) to print the file. After printing, the temporary text file is automatically deleted from the disk. This is mentioned only because if you select a large amount of data, there may be a noticeable delay before printing begins while Red Ryder collects data into the temporary text file.

The Append Data To Archive File Command

Under the **Edit** menu, the **Append Data To Archive File** command either creates a new file called "Archived Screens" if one doesn't exist, or appends to the end of an existing "Archived Screens" file the selected text.

Each time Red Ryder starts up, it will use the same disk (and HFS folder) that Red Ryder is located in for the "Archived Screens" file. You can change the location of the "Archived Screens" file that Red Ryder will create or append to by choosing the **Set Archive Volume...** command under the **File** menu. That volume (or HFS folder) will then be used until it is either changed again, or you quit Red Ryder.

The "Archived Screens" file is a plain text file, which means that it can be imported into any Macintosh text editor or word processor (see the user manual for your text editor or word processor for details on loading plain text files).

There are two menu commands for appending text to the "Archived Screens" file. Choosing **Archive Display Screen** from under the **Local** menu will save only those lines of text displayed in the Terminal Display Window. If you wish to save all of the text in the Saved Screens Buffer, choose **Archive All Screens** from under the **Local** menu. This command will not save any blank lines in the saved screens buffer until it comes to the first non-empty line in the buffer.

The Paste Command

Under the **Edit** menu, the **Paste** command will send the contents of the Clipboard to the serial port. What is actually happening here is that Red Ryder creates a temporary text file and copies the text in the Clipboard into this file. It then uses the **Send Text File...** menu choice to send the data to the serial

port (this command is discussed in more detail in the "Getting It From There To Here Or Here To There" chapter). Once the file is sent (or you choose "Cancel File Send" from under the File menu, the temporary text file is destroyed.

The Undo And Clear Commands

Under the **Edit** menu, the **Undo** and **Clear** commands are not used by **Red** Ryder. They are there to support those desk accessories which make use of these commands.

Choosing The Font Text Is Displayed With

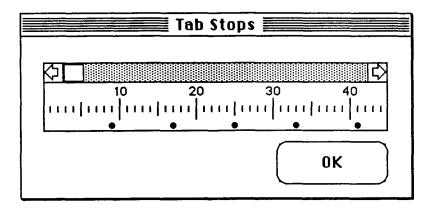
You can choose which font and point size text in the Text Display Area is displayed by choosing **Set Font** under the **Local** menu. Red Ryder will let you choose any installed font when you are emulating a TTY terminal, but when you are emulating either a VT52 or VT100 terminal, you may use only the TTY-VT52-VT100 font (although you may select either 9 or 12 point size for this font).

Red Ryder will allow you to use only font sizes that are installed in the System Folder of the startup disk.

Tab Stops

When in the course of displaying text Red Ryder receives a tab control character, it will interpret this to mean "print spaces up to the next tab stop or the end of the current line." By choosing **Tab Stops...** from under the **Local** menu, you can specify at which columns to place tab stops.

When this command is chosen, a dialog box as shown below is displayed on your screen:



The dialog box contains a column ruler, with small black circles showing where tab stops are located. To set a tab stop, just point your mouse at the desired column in the ruler and click - a small black circle should appear. To remove a tab stop, just point at the column in the ruler containing a small black circle and click. The horizontal scroll bar above the ruler can be used to show columns not currently displayed in the ruler.

There are two important things to remember about tab stops. First, just because you place your tab stops in certain positions does not mean that the remote machine has agreed (or even knows about) these positions. It's similar to taking a typed report from one typewriter in your office and putting it in a second typewriter. If you are capturing (or archiving) the text on the screen, you will probably have to set the tab stops again once the file is loaded into a text editor or word processor.

Secondly, Red Ryder's tab stop ruler is based on columns, whereas many word processors base their tab stops on inches (or centimeters). Therefore, if you are using a proportionally spaced font (where some characters are wider than others), don't be alarmed if columnar data doesn't line up on the Red Ryder screen. If you will be receiving columnar data, I suggest you use a monospaced font (where all characters are of equal width), like the Monaco or TTY-VT52-VT100 fonts.

Serial Port Settings

A modem is a device that works with a computer in the same manner as a telephone handset works with your head. Both of these translate the original communications media (for humans: sound, for computers, digital electronic signals) into the bleeps and blorps that the telephone lines are capable of carrying. The receiving handset or modem then retranslates the bleeps and blorps back into voice or electronic signals.

Consider a telephone conversation between people. Both persons must (either implicitly or literally) agree to speak in a language both participants understand, and a certain degree of etiquette (or protocol) must be followed throughout the conversation (such as not hollering wildly at the same time). Using computers and modems is very much the same.

The first thing you must know before calling another computer is what language it expects you to use. For computers, we refer to this language as the <u>communications protocol</u>. If you set up your computer to use a different protocol than the computer you call uses, you'll either get "garbage" characters on your screen or nothing at all. Sooner or later, one of you will get frustrated and hang up.

Communications protocol can be divided into several components. We call each component a <u>parameter</u>. Each parameter must be set up correctly to communicate properly.

Baud Rate - this is how quickly the computers are capable of "speaking" to each other. The choices Red Ryder offers are 300, 450, 1200, 2400, 4800, 9600, 19200, and 57600 baud.

Parity - This is a simple form of error checking. Red Ryder supports NO Parity, ODD Parity, EVEN Parity, MARK Parity, and SPACE Parity.

Data Bits - This size of each character. Red supports 5, 6, 7, and 8 data bits.

Stop Bits - Additional signals sent at the end of each character transmitted. Red supports 1, 1.5, and 2 stop bits.

Duplex - This is the "etiquette" we spoke about earlier. Red supports FULL Duplex, HALF Duplex, ECHO Duplex, and NULL Duplex. HALF Duplex is what some users refer to as "local echo", and should not be confused with ECHO Duplex. ECHO Duplex will never be used by both callers at the same time, and is only used by Red Ryder when it is acting in a host capacity (see the chapter on the Host Mode for more details).

Duplex controls what Red Ryder does with the characters you send. When you are using FULL Duplex, the characters are sent directly through the serial port.

FULL Duplex is used when the remote machine will echo back the characters you type. When you are using HALF Duplex, Red Ryder not only sends the characters you type through the serial port, but also to your display screen. This is used when the remote machine does not echo back the characters you type.

ECHO and NULL Duplex are provided for special cases. ECHO Duplex functions like HALF Duplex, except that it also echoes back any characters received (much like a remote machine would when you are using FULL duplex). ECHO duplex is provided for when you are using Red Ryder in a host capacity (see the chapter "Host Mode" for details). NULL Duplex also acts like HALF Duplex by sending the characters you type to your display screen, but it does <u>not</u> send those characters over the serial port. NULL Duplex is used mainly for local testing purposes.

You'll need to contact the operators of the computer you'll be calling (or consult the documentation that describes the service) to find out how to set up these parameters. There will be times, however, that this is not practical or possible. In this case, you'll have to resort to an educated guess. This isn't as hard as it sounds.

I recommend that the first time you call that you use 300 baud. It is the lowest common denominator, as well as the slowest standard speed used. Once connected (or "online", as we in "the biz" say), you can find out if the service supports a faster baud rate.

Next, let's talk about duplex. Most professional computer services and bulletin board services (BBS's) will echo back the characters you type so they'll display on your screen. For this reason, start out with FULL duplex. If you can't see what you're typing, switch to HALF duplex. If you are using HALF duplex and see double characters ("HHEELLLLOO"), switch to FULL duplex.

Pretty easy so far, eh? This leaves only 3 parameters (Parity, Data Bits, and Stop Bits) with "billions and billions" of combinations. Not to worry, only a few combinations are used by 99% of the systems out there. The other 1%, of course, were put on this earth just to frustrate documentation writers who try to make things easy for beginners.

The first time you call a system, use NO Parity and 8 Data Bits. If you get "garbage" characters or the other computer does not respond at all, wait a few seconds and then type the "RETURN" key a few times and see if that helps. If you still have no luck, hang up, set your parameters to EVEN Parity and 7 Data Bits, and call back. Most likely, one of these two settings will work. If not, you're out of luck as far as an educated guess is concerned, and will have to get the information from the operators of the system. You will probably never need to fool with Stop Bits, just set it to "1" and forget about it.

Red Ryder does not "know" what, if anything, is connected to the serial port. Whatever you type at the keyboard is sent through the serial port. If characters

are received through the serial port, they are displayed as appropriate in the terminal window. Therefore, you can send your modem's particular commands (such as for auto-dialing) just by typing directly to the modem. Because Red Ryder is designed to support any modem that can be hooked up to a Macintosh, we do not discuss in detail the methods that are specific to various modems for such things as dialing and hanging up in this document. You do need to read through the manual that came with your modem to exploit its capabilities with Red Ryder.

The vast majority of Red Ryder will be using "Hayes compatible" modems that support the Hayes Smartmodem "AT" command set at least to some degree - we have found that some manufacturers (including at least...ahem...one major computer corporation) have used the term "compatible" rather liberally. The manual that came with your modem will go into more detail, but we can serve up a few basic pointers here to get you started.

Sending Commands To The Modem

Before sending any dialing commands to the modem, you should make sure the modem has recognized the serial port settings you are using. To do this, type the letters <u>AT</u> and press the Return key. Some modems will let you get away with lowercase letters, others demand all capital letters.

If everything goes well, the modem will respond by sending the word "OK" to Red Ryder, which will be displayed on the screen. If it doesn't, try sending the <u>AT</u> command a couple of more times before giving up.

The <u>AT</u> command is used to get the modem's attention. It doesn't have to be sent before every modem command, but it should be used immediately after starting up Red Ryder, and each time after you change the serial port settings.

Dialing A Number

If you are using a touch-tone telephone, the modem dialing command <u>ATDT</u> is used. If you are using a pulse telephone, the modem dialing command <u>ATDP</u> is used. Follow the dialing command with the number you wish to dial, like this:

ATDT 555-1212 (press Return)

The hyphen in the above command is ignored by the modem. You can introduce a pause during dialing by inserting a comma in the number to be dialed. The dialing command:

ATDT 9,555-1212

would dial a 9, wait 2 seconds, then dial 555-1212. Pauses are useful when you need to get an outside line through a switchboard. You can string more than one comma together for longer pause durations (in increments of two seconds).

Hanging Up The Modem

When your modem connects with another modem, it goes into a characters pass-through mode, which means that it leaves the command mode and passes through the characters you type to the other modem. To get back in command mode in order to disconnect, you need to:

1) Wait at least three seconds since the last character you typed.

2) Type three plus signs quickly (don't press Return).

- 3) Wait a few seconds and the modem should come back with the message "OK".
- 4) You're now in the command mode. Type the modern command <u>ATH</u> and press Return. The modern should disconnect from the phone line.

Telling Your Modem To Answer An Incoming Call

The command ATS0=1 (that's a zero) is used to tell the modem to answer an incoming call on the first ring. The command ATS0=0 tells the modem never to answer the phone. If you forget to set the modem to auto-answer mode and have an incoming call (you get "RING" messages), you can tell the modem to immediately take the phone line off hook and attempt to connect with the caller's modem with the command ATA.

Armed with your modem manual and the information in this chapter, you should have all you need to try and connect with a remote machine for the first time.

The DTR Modem Line

On the newer Macintosh models with the round 8-pin serial port connector, a special output handshake line is provided. This line is typically connected to the RS-232 signal line designated as DTR (Data Terminal Ready).

When the signal on the DTR line is "dropped", many modems will automatically disconnect (hang up) the phone line. Local Area Networks may use the DTR line to signal connection.

In the dialog box presented by the Change Serial Port Settings... command, if the choice Don't drop DTR when exiting Red Ryder is selected, the connection will not be dropped when you exit Red Ryder, and you should be able to return online by executing Red Ryder again. If this choice is not selected, the DTR line will be dropped when you quit Red Ryder.

If you click in the **Drop DTR For 1 Second** button, the DTR line will be dropped for one second, which may be useful for signaling a Local Area Network or modem equipped PBX system.

Neither of these settings will have any effect unless the RS-232 DTR line is connected to the output handshake pin of the serial port Red Ryder is communicating through, or if the device connected to the serial port is instructed to ignore the state of the DTR line.

The General Status Bar

To select the General Status Bar display, choose **General Status Bar from** under the **Local** menu. The General Status Bar is composed of six different control areas. From left to right:

• The Elapsed Time And Billing Clocks:

00:47:26 \$3.91

• The Display Screen To Printer button:



• The Display Screen To "Archived Screens" File button:



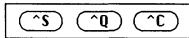
• The serial port settings display:

Serial Port Settings 1200-N-8-1-FULL

The Pause/Resume Remote Button:

Pause Remote

The Control Character Buttons:



The Elapsed Time And Billing Clocks

The Elapsed Time And Billing Clocks are used to keep tabs on how long you've been connected, and how much money you've spent so far. To reset the clocks back to 00:00:00 and \$0.00, click in the box containing the clocks.

You can calibrate the billing clock to the desired charge by selecting **Set Billing Cost..** from under the **Service** menu. This will display a dialog box prompting you to enter the cost per minutes in 100ths of a cent. In other words, if your service charges you \$5.00 per hour, you would divide 500 cents by 60 minutes to get 8.33 cents per minute. Divide 8.33 cents by 100 to get 833, which is your cost per minute in 100ths of a cent. This may seem a little convoluted at first, but by specifying the cost per minute in 100ths of a cent, the Billing Clock offers a reasonable degree of precision.

The Display Screen To Printer Button

Clicking in the Display Screen To Printer Button will send the lines of text in the Terminal Display Window to the printer. For details in setting up and using a printer with Red Ryder, see the chapter "Using A Printer With Red Ryder".

The Send Screen To "Archived Screens" File Button

Clicking in the Send Screen To "Archived Screens" File button is equivalent to choosing the **Archive Display Screen** choice under the **Local** menu. This menu command is described in detail in the chapter "The Terminal Display Window".

The Serial Port Settings Display

The Serial Port Settings Display shows you at a glance what baud rate, parity, databits, stopbits, and duplex is being used by the serial port Red Ryder communicates through. Clicking in this box is equivalent to choosing **Change Serial Port Settings...** from under the **Local** menu. Serial port settings are discussed in detail in the chapter "Serial Port Settings".

The Pause/Resume Remote Button

The Pause/Resume Remote Button is actually two buttons, "Pause Remote" and "Resume Remote" which toggle back and forth each time you click on them. Clicking the "Pause Remote" button will send a CTRL-S (also known as XOFF) control character, which many host machines understand to mean "stop transmitting".

Clicking on the "Resume Remote" button will send a CTRL-Q (also known as XON) control character, which many host machines understand to mean "resume transmitting".

The use of CTRL-S and CTRL-Q to start and stop transmission is known as XOFF/XON handshaking. It's not guaranteed that every host machine you'll connect with will support XON/XOFF handshaking, but the vast majority should. Let's say you are connected to a host at 1200 baud and it is displaying some text you are interested in. Unless you're a much faster reader than the average bear, text will probably start scrolling off the top of your screen before you can finish reading it. When this happens, click on the "Pause Remote" button and the remote machine should stop transmitting. When you're caught up, click on the "Resume Remote" button and the remote machine should resume transmitting. This button makes it easy to start and stop the host by positioning the mouse over the Pause/Resume Remote Button and clicking your mouse to throttle the transmission to a more comfortable speed.

The Control Character Buttons

The Control Character Buttons can be modified to send any control character you wish by selecting **Edit Control Buttons...** from under the **Edit** menu. These buttons simply send the displayed control character when you click your mouse in them. They are useful for when your hand is on the mouse rather than the keyboard, or when certain control characters are needed but not readily available on the keyboard.

Keyboard Specifics

Red Ryder uses the keyboard in such a manner to support all of the functions commonly used in telecommunications. Red Ryder will automatically adjust to either of two keyboards, the standard keyboard, which was shipped with the original 128K and 512K Macintoshes and could be configured with an optional numeric keypad, and the Macintosh Plus keyboard, which was shipped with the Macintosh Plus and has a built in numeric keypad.

These two keyboards represent the vast majority of those in use at the time of Red Ryder's release. However, I am certain that radically different keyboards will be shipped for use with newer Macintosh models. This will have an affect only on the VT100 keypad emulation, and a special "Install VT100 Keypad" utility program has been created in anticipation of physical alterations to the keyboard.

Red Ryder has the capability to support many different needs as far as special keys are concerned. Because there several keys found on other computers and terminals that are not found on the standard and Macintosh Plus keyboards, Red Ryder gives you the ability to "map" the keyboard to your special needs. Browse through this information right now so that you know it exists. Chances are you may never have to adjust Red Ryder's default settings unless you use VT100 or VT52 terminal emulation, or communicate with a host that falls into the "oddball" category.

In the following descriptions, we will refer to the key with the cloverleaf symbol on it as the "Command" key.

Under the Customize menu, there is a Keyboard Mapping Preferences choice that brings up a dialog box with the following items:

key sends "x" (where "x" is a value from 0 to 255) makes the right pointing accent (which is at the top left corner of the keyboard) send whatever character you wish. For instance, to make this key send an ESCAPE, you would enter the value 27 (which is the ASCII code number for ESCAPE). Note that there is not a way (except through a macro key) to send a right pointing accent character when this option is selected.

If the **Backspace key is DEL key** choice is enabled, the Backspace key will send a DELETE control character (ASCII code 127). Otherwise it will send a normal backspace control character (ASCII code 8). If your Backspace key doesn't seem to work right, check the state of this choice.

If the Modem BREAK signal keys supported choice is enabled, the keyboard combination Shift-Backspace will send a short (233 millisecond) modem BREAK signal (if your modem supports it). The keyboard sequence

Shift-Command-Backspace will send a long (3 1/2 - 4 second) modern BREAK signal.

If the VT100 cursor key diamond supported is enabled, you can simulate the VT52/100 cursor keys with the following Macintosh keyboard combinations:

Macintosh Keyboard Action Command-Backspace

Command-Return

Command-\

Command-]

Sends:

Same as VT100/52 "up"

cursor key.

Same as VT100/52 "down"

cursor key.

Same as VT100/52 "right"

cursor key.

Same as VT100/52 "left"

cursor key.

This feature will probably be only useful with the standard Macintosh keyboard which does not have a numeric keypad attached. This gives you a set of cursor keys on the keyboard arranged in a diamond shape.

The Control characters are sent with choice allows you to choose one of two keys that can be used to send control characters. In addition to the normal A-Z alphabet and punctuation characters we use in normal written communications, there are many additional characters significant in communicating by computer. What they are and how they are used is really up to each individual service you'll connect with. Many are standard, but don't bet your last dollar that any or all will be recognized until you've read the service's documentation or have had a chance to experiment. Often, the correct characters to press will be displayed on your screen by the remote service at various times, like:

Press CTRL-S to pause, CTRL-Q to resume, or CTRL-C to quit.

The "CTRL", by the way, is a common abbreviation for "control". A control character is used for various commands to the remote service. A 'C' and a CTRL-C are two different characters and should not be confused or used interchangeably.

Since the early Macintosh keyboards were not provided with a "Control" key, an interesting dilemma confronted me while writing Red Ryder. The Command (cloverleaf symbol) key was a logical replacement for the Control key, but the user would have to give up Command key menu choice equivalents (which many users favor over the mouse in certain situations). If the OPTION key were used international characters could not be directly typed.

Perhaps it is wishy-washy, but it has been decided that the decision should be made by the user, on the basis of personal preference and need. Therefore, the **Keyboard Mapping Preferences** dialog box allows you to decide whether

to use the Option key or the Command key to send control characters. If the Command key is chosen, the menu/key equivalents are not displayed or supported in Red Ryder's menus. Otherwise, they appear in the menus and can be used as shortcuts for menu choices. The NULL control character (ASCII code 0) can be sent by holding down your designated Control key (Option or Command) and pressing the Spacebar key. To send a control character (CONTROL-A to CONTROL-Z is ASCII code 1 to ASCII code 26), you hold down your designated CONTROL key and type the desired letter key.

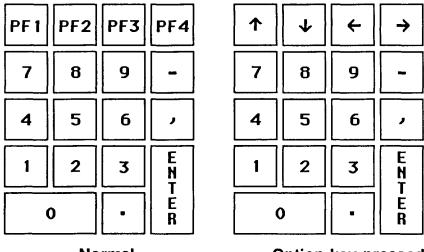
The Command and Option keys are used like a Shift key. To send a CONTROL-A, you would press and continue to hold down the designated Control key and then type the character "A".

In this documentation, and in some character strings used by Red Ryder, we designate a control character by preceding it with a caret (^) symbol. Some of the control characters you're likely to run across frequently are:

ASCII Code	<u>Kev</u>	Typical Use
3	<u>Key</u> ^C	Usually means "interrupt this function"
7	^G	Sound a "BELL" or beep
8	^H	Same as your Backspace key
9	^	Same as your Tab key
10	^J	A linefeed character
12	^L	Clear the screen or form-feed the printer
13	^M	Same as your Return key
17	^ Q	XON character, tells other computer to resume transmission after an XOFF.
19	^ S	XOFF character, tells other computer to halt transmission until an XON is received.

The **Return key sends:** option lets you choose whether the Return key sends just a carriage return, or a carriage return followed with a linefeed character. If the characters you type appear perfectly on your screen but the host system does not respond when you press the Return key, or lines seem to overprint each other when you press Return, chances are the remote system probably requires an additional character after the carriage return called a linefeed character. Don't select this option indiscriminantly, some host systems will <u>not</u> function correctly if this option is turned on.

If you have an Apple numeric keypad connected to your standard Macintosh keyboard, or a Macintosh+ keyboard with the numeric keypad built in, its keys will function exactly like those on a VT100 keypad when VT100 or VT52 emulation is active. The keys are mapped like this:



Normal

Option key pressed

Because many VT100 applications map special keys to the above <u>physical</u> arrangement, it was my decision to map the VT100 keypad in its original alignment, rather than according to the characters physically printed on the Macintosh key caps.

If a Macintosh Plus keyboard is used, the cursor keys at the bottom right corner of the keyboard are automatically supported.

If you are using a non-U.S. keyboard, or are using a newer keyboard than that supplied with the Macintosh Plus (like the ones shipped with Macintosh SE and Macintosh II machines), you may find that some of the special VT100 keys don't work as documented above. Don't despair, you can adapt Red Ryder to any keyboard once you've read the chapter "Redefining The Standard VT100 Special Keys".

The Buffered Keyboard Status Bar

The Buffered Keyboard Status Bar was added mainly for use with the real time conferencing features of several commercial timesharing services like CompuServe, GEnie, and Delphi. A real time conference is where two or more (sometimes several dozen or even hundred) people get together and type to each other in a CB-radio fashion, exchanging opinions, tips, bad jokes, etc. The value of the Buffered Keyboard Status Bar is that it gives you the opportunity to correct mistakes (or obscene punch lines) before sending them, and it sends the characters typed all at once, so that even the slowest typist doesn't hamper the faster typists in the crowd. If you take advantage of real time conferences (I suggest you do - they can be very educational and a <u>lot</u> of fun), you'll find the Buffered Keyboard Status Bar indispensable.

To display the Buffered Keyboard Status Bar, select **Buffered Keyboard Status Bar** from under the **Local** menu. Whenever the Buffered Keyboard Status Bar is displayed, the characters that you type are not sent to the serial port immediately, but are instead held in an editable buffer until the Return key is pressed. When that happens, the characters are "flushed" out of the buffer through the serial port all at once.

The exception to this is that all control characters, except tabs and carriage returns, and characters from sources other than the keyboard (such as Macro Key Strings, VT100 Status Bar keys, and Procedure TYPE commands) are sent immediately through the serial port without effect to the buffer. It is not possible to put control characters in the buffer. Tabs are expanded as spaces (according to where you have tab stops set) just as inside the Text Display Area.

The text inside the buffer is editable, which means it can be selected with the mouse, and then cleared, copied, or cut to the Clipboard. To use the Cut, Copy, Paste, and Clear commands under the Edit menu with the buffer, hold down the Option key when selecting the desired command from under the Edit menu. The command must be selected from the Edit menu with the mouse - keyboard menu command equivalents won't work for this.

For example, holding down the Option key and selecting **Paste** from under the **Edit** menu tells Red Ryder to paste the contents of the Clipboard to the buffer, rather than directly to the serial port.

Some services have a limit of how many characters you can type before sending a carriage return. Exceed that line length limit, and the service will probably choke, send you an obscure error message, and throw away what you typed into oblivion. For this reason, Red Ryder supports a modifiable right margin for the Buffered Keyboard Status Bar that it won't let you type past. To change the position of this margin, the dialog box brought up by the **Personal**

Preferences choice under the Customize menu contains the choice Buffered keyboard right margin limit. If the right margin is set to greater than 80 columns, the horizontal scroll bar above the buffer is activated to allow you to scroll left and right for viewing all of the characters in the buffer.

You'll also notice that to the left of the horizontal scroll bar, your current right margin column and the number of characters in the buffer is displayed. As you type past the right edge of the buffer, the buffer is automatically scrolled left for you so you can keep your eyes on what you're typing. Once you hit the right margin, a bell is sounded and no more characters will be accepted until you either delete one or more characters, or send the contents of the buffer by pressing the Return key.

The characters in the buffer are always displayed in Red Ryder's TTY-VT52-VT100 font at 9 point size.

The Macros Status Bar

The Macros Status Bar is displayed by selecting Macros Status Bar from under the Local menu. It is composed of three control areas, from left to right:

• The Macro Key Buttons:



• The Set Selection And Display Control:

실씨닷

• The Elapsed Time And Billing Clocks:

01:42:49 \$8.49

The Elapsed Time And Billing Clocks

The Elapsed Time And Billing Clocks operate exactly like their counterparts in the General Status Bar, and are described in the chapter "The General Status Bar".

What Is A Macro Key?

Once you've started using various services regularly, you'll find that you'll be typing many of the same words and commands over and over. Macro Keys are used to send up to 40 characters of your choice with only one keystroke or mouse click, or they can be used to quickly execute a compiled Procedure File. There are 30 Macro Keys, divided into 3 Sets of ten Macro Keys each, which we refer to as Set #1, Set #2, and Set #3. Each of the three Sets contains Macro Keys numbered 0 to 9. A Set containing a Macro Key must be active before that Macro Key can be executed. You know that a Macro Key Set is active when its Set number is shown in the middle of the Set Selection And Display Control - in the above illustration, Set #2 is active. To make a different set active, click in the arrows in the Set Selection And Display Control. The new active set will scroll into view and its Set number will be displayed in the middle of the Set Selection And Display Control.

In the Macros Status Bar, the top row of Macro Key Buttons contains buttons #0, #1, #2, #3, and #4. The bottom row contains Macro Key Buttons #5, #6, #7, #8, and #9.

To execute a macro key, you hold down the COMMAND key and type a numeral from 0 to 9 corresponding to the Macro Key you wish to send (i.e. type COMMAND-1 to execute Macro Key #1 in the currently active Set), or click in the corresponding Macro Key Button in the Macros Status Bar.

Each Macro Key consists of two different strings of characters, the macro key

label and the macro key string. The label is used to remind you what the Macro Key does. Macro Keys become visible in the Macros Status Bar when they have a label. If the label is empty, the Macro Key Button is not displayed in the Macros Status Bar and is not selectable with the mouse.

To store characters inside a macro key label or string, select the **Edit Macro Keys** choice from under the **Edit** menu. You'll first see a dialog box asking you to specify which set and macro key number in that set to edit. Clicking in the **Erase All** button will erase all Macro Key Labels and Macro Key Strings. After choosing the set and key number, you'll see a new dialog box with two editable text boxes. The top box is titled "String:" and it contains the actual characters you wish that macro key to send, or the file name of the compiled Procedure File to execute. The bottom box is titled "Label:", and it contains a "name" you wish to give to that key. The label can be up to 13 characters long, and the characters are shown centered in the Macro Key Button.

A carriage return character is not automatically sent at the end of a macro key string. However, it just so happens that there is a way to embed control characters in a macro key string, and there is a certain control character that is the same as what your RETURN key sends. This control character is CTRL-M. You specify control characters by preceding it with a caret character, like ^M for CTRL-M. To send a caret character, type in two carets side by side (^^).

Here's an unlikely (but illustrative) example:

HELLO'ETHERE'BOB'M

This example would send the characters "HELLO", followed by a CTRL-E, followed by the characters "THERE" followed by a single caret character, followed by the characters "BOB", followed by a carriage return (which is the same a CTRL-M).

If the above makes perfect sense, you're in good shape. Otherwise, you need to go back and review the last few paragraphs until it is clearer.

For the most part, the only control character you'll probably need to ever use in a Macro Key is CTRL-M for carriage returns. Just remember that:

HELLO THERE^M

would send a carriage return at the end of "HELLO THERE", and:

HELLO THERE

would not. For those of you who have found the need to send linefeeds after carriage returns, your "magic" sequence becomes:

HELLOTHEREYMYJ

Macro Keys are useful for holding passwords, commonly used commands, and even quick and dirty modem dialing commands. If you have a Hayes compatible modem you could set up a Macro Key that contains:

ATDT 555-1212^M

When you sent that Macro Key, it would issue the appropriate dialing command to dial touch-tone the number 555-1212. Please change that number before experimenting or you'll have a Directory Assistance operator who thinks they have a "breather" on the line while you're busy celebrating success.

Remember that each Macro Key String can only hold 40 characters, including any caret characters. If you try to enter more than 40, the surplus will be "lopped off" into oblivion. So there.

Later on you'll learn about a programming language contained in Red Ryder which can be used to write something called a Procedure. Procedure Files are used to completely automate a portion of, or even an entire online session. If the first character of a Macro Key String is a backslash (which is <u>not</u> to be confused with the slash ("/") character) Red Ryder will use the rest of the characters in the Macro Key String to be the file name of a Procedure file you wish to execute immediately. No characters are sent through the serial port by the Macro Key itself when used in this manner. This is a much faster way of executing frequently used Procedures than going through the **Initiate Procedure...** menu command and then navigating through disks and folders to find the Procedure File you wish to run.

In other words, if the Procedure File named CALL COMPUSERVE is located on the disk named PROCEDURES, you would need to set up the Macro Key String to read:

\PROCEDURES:CALL COMPUSERVE

The colon character is what separates the volume and file names.

One of the most common problems in using Macro Keys to execute Procedure Files is getting the file name of the Procedure File correct. Bad spelling aside, the Macintosh allows you to have leading and trailing spaces in front of disk, folder, and file names, and these spaces <u>must</u> be present in the Macro Key String or else you'll get a "Procedure File Not Found" error. In other words, unless you have X-ray vision, it can be <u>real</u> difficult to get such file names correct. For this reason, instead of typing in a file name, use the **Get Procedure File** button provided in the dialog box. When clicked, you will be presented with a standard file selection dialog box and prompted to select a Procedure File. Once a Procedure File is selected, Red Ryder will construct a valid path name to the file and insert it in the Macro Key String (complete with

the leading backslash character).

You'll soon find that even 30 Macro Keys just aren't enough for your evergrowing sophisticated needs. Red Ryder allows you to save your Macro Keys in a disk file to be loaded in at a later date. Simply choose the appropriate choice Save Macro Keys To Disk or Load Macro Keys From Disk from under the Customize menu to accomplish this.

You can also use macro keys to set up special escape-code command sequences used by VT52 and VT100 applications. To send an ESCAPE control character, use the sequence "^[" (caret-left square bracket).

Red Ryder can also accept certain other sequences of characters in a Macro Key String to perform special functions:

Sequence	<u>Function</u>
^\$	Sends a DEL character (ASCII code 127)
^!	Sends a short modem break signal
^@	Sends a long modem break signal
^#	Drops the DTR serial port line for 1 second

These sequences don't have to appear alone, they can be embedded within a Macro Key String's characters.

The VT100 Status Bar

The VT100 Status Bar is used for two purposes:

- To display the condition of the four LED's (small lights) found on an actual VT100 terminal, but not on a Macintosh.
- To allow Macintosh owners that do not have a numeric keypad to simulate PF keys, cursor movement keys, and numeric keypad keys found on actual VT52 and VT100 terminals.

The VT100 Status Bar is only used when you are emulating a VT52 or VT100 terminal (see the chapter "Terminal Emulation Specifics" for more details). It has no function when you are emulating a TTY terminal. To display the VT100 Status Bar, choose VT100 Status Bar from under the Local menu. The VT100 Status Bar is divided into three control sections:

• The PF keys and LED display:

PF1	PF2	PF3	PF4
L1	L2	L3	L4
			1

- The cursor movement keys:
- The numeric keypad keys:

		+		·		
0	1	2	3	4	_	,
5	6	7	8	9		ENTER

With the exception of the four LED's and the two blank boxes surrounding the up cursor movement key, clicking your mouse inside any of the boxes is equivalent to pressing the same key on an actual VT52 or VT100 terminal. If the VT100 Auto-repeat mode is set (VT100 modes are discussed in the "Terminal Emulation Specifics" chapter), the keys in the VT100 Status bar will repeat as long as the mouse button is held down inside them.

The two blank boxes surrounding the up cursor movement key are not used and will just beep at you if you try to select them.

The four LED's L1, L2, L3, and L4 will be shown as white characters on a black background when they are "lit". In the above illustration, L2 and L4 are "lit", while L1 and L3 are not "lit".

Redefining The Standard VT100 Special Keys

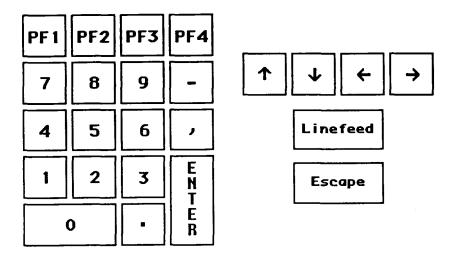
As discussed in the chapter "Keyboard Specifics", the VT100 special keys not found on all Macintosh keyboards (specifically, the cursor movement keys, PF1 through PF4 keys, and numeric keypad keys) are automatically mapped for the keyboards shipped with the original Macintosh 512K and Macintosh Plus computers.

It has become apparent that Apple will ship at least two additional keyboards for use with the Macintosh SE and Macintosh II machines, and it would be unwise not to anticipate down the road that there will be more keyboard variations than this author has bad habits (those who have met me in person will attest that my ahem...eccentricities... can be counted only with the help of scientific notation). This, coupled with the fact that there already exist several international keyboards that have different physical (and sometimes logical) key arrangements led me to address this problem with an all-purpose special key remapping utility. This utility, found on the Red Ryder master disk, has an icon that looks like this:



Please recognize that this utility is provided only for the special cases that require it. Those who are using Apple keyboards shipped with the Macintosh 512K or Macintosh Plus will not normally need to use this program.

The <u>Install Keypad</u> utility is used to install special key mapping information in a Settings File. When this application is first executed, you will be prompted to select a Settings File to modify with the new key mapping information. Once you have done this, the a dialog box is displayed that contains the following graphic depiction of the special keys that can be remapped:



These keys are grouped for illustration purposes as they are grouped on an actual VT100 keyboard. You can choose to assign their functions to anywhere on the keyboard. To remap a key, just click your mouse button with the cursor pointed inside the key to be remapped in the picture. A new dialog box will appear asking you to type the key on your keyboard that will correspond to the remapped key in the picture. As you type this key, remember that the Option, Shift, and cloverleaf keys <u>are</u> significant, and will have to be supplied in the correct combination ("Shift-Option-Cloverleaf-X" would be a proper key definition) later to get this key function.

The following is important so please read it carefully. When you remap keys using the "Install Keypad" utility, it does not make any changes to the chosen Settings File <u>until</u> you click on the **Install Special Keys** button. Therefore, if you go through and specify a complete redefinition of these keys and the click on the **Quit** button without first clicking on the **Install Special Keys** button, none of your changes will actually be saved in the Settings File.

If you later change your mind about the remapping and want Red Ryder to go back to its automatic keyboard recognition (of the earlier Macintosh keyboard models, that is), just start up the "Install Keypad" utility, choose the Settings File that you earlier modified with this utility, click on the **Remove Special Keys** button, and then click on the **Quit** button.

The second important thing to remember is that whenever you click on the **Install Special Keys** button, it completely replaces all of the keys in the picture with the definitions you have stored. This means that if you run this application, remap only the cursor movement keys, and then click on the **Install Special Keys** button, the changes will be stored for <u>all</u> of the keys pictured. Until you define a key, it is given an "impossible" keyboard mapping. So, remember to remap all of the pictured keys with this utility, not just one or two, or you'll lose the ability to send the other pictured keys.

The "Install Keypad" utility has a built in feature to help you keep track of which

keys have been given new definitions, and which still remain in limbo. Once you have clicked on a key in the picture and typed its new keyboard mapping, it will be displayed as a gray square in the special keys picture. The undefined keys will be shown with a white background.

I make no apologies about the fact that the "Install Keypad" utility was a 23rd hour necessity brought on by machines I did not have access to during this software's development. As such, I recognize that some improvements are immediately recognizable and will be addressed in future versions of Red Ryder. One of these is the fact that the **Show Keypad Map** button in the dialog box displayed by the **VT100 Modes...** menu choice is "blind" to changes made by the "Install Keypad" utility and may not reflect the physical remapping you have chosen. Therefore, I recommend that you write down the remapping of any keys that vary from the normal and keep that list handy for reference.

It is a fact of software development that certain design considerations of a non-aesthetic nature (translation: it ain't pretty, but it works) are sometimes unavoidable if one is to assure a reasonable longevity for a piece of work. One does not decree the bizarre ruminations that may come out of Apple's keyboard engineering labs - one merely supplies the best profanity of their choice and sidesteps them.

However, it may just so happen that this utility, in all of its user-ugliness, may provide a few additional solutions unintentionally. An example of this is a conversation I had with a fellow concerning a keyboard need he had for a particular mainframe/VT100 application he was tied to. The application in question used the "Linefeed" key almost exclusively, and the "Return" key hardly ever. His complaint therefore was that typing a CTRL-J (a linefeed) was awkward, and he really wished to remap the "Return" key as a linefeed key on his Macintosh Plus keyboard. After writing this utility, I found that this could easily be achieved with the "Install Keypad" utility. The icing on the cake is that since the Macintosh Plus keyboard defines a "Shift-Return" as the same thing as an unshifted "Return" keypress, he could remap the unshifted "Return" key as the "Linefeed" key, and then use "Shift-Return" for the seldom needed carriage return.

Just remember that if you do something like this (just changing the function of one key), don't forget to remap all of the other pictured keys to their original mapping.

Customizing The Terminal Emulation

Red Ryder has the capability to emulate several standard computer terminals, including a generic TTY (Teletype or like an electric typewriter) terminal, a Digital Equipment Corporation VT52 terminal, and a Digital Equipment Corporation VT100 terminal. While this may not make Red Ryder perform all things for all people, it does satisfy the requirements for the vast majority of general telecommunications applications.

Under the Customize menu, the Terminal Emulation Preferences choice presents a dialog box containing the following options:

• Display columns: Most terminals (and telecommunications programs) limit you to a small number of columns per line choices. Red Ryder allows you to choose any number of columns between 20 and 132 (inclusive). Some hosts will expect a standard line length (typically 40, 80, or 132 columns) and will format its sent text to reflect that length. However, if you are using an especially large font size and don't care much for a lot of constant horizontal scrolling, you may wish to reduce the line length.

The size of the line length does not affect incoming text that is captured with the **Capture Incoming Data To TEXT File...** menu choice, which is to say that no additional carriage returns are placed in the file than what were sent by the remote machine. It will, however, affect where carriage returns are placed when you select a block of text using the mouse and then copy it to the Clipboard file using one of the commands under the **Edit** menu.

- TTY Full-Screen Clears: As explained in more detail below, one of the control codes the TTY emulation supports is the "clear screen" (ASCII Code 12) command. When this control code is received, Red Ryder will clear the last 24 lines of text received in the Terminal Display Window. You may, however, wish to keep that text around, in which case you should checkmark the Ignore TTY full-screen clears option so that these commands are ignored.
- •Cursor style: This option does not affect the mouse cursor, but the small text cursor which shows you where the next received character will be displayed in the Terminal Display Window. Some people prefer a large block shaped cursor, in which case the **Block** option should be checkmarked, others prefer an underline shaped cursor, in which case the **Block** option should not be checkmarked. Additionally, you can decide whether you wish the cursor to flash constantly or remain solid by checking or not checking the **Flashing** option.
- Terminal emulation: This is where you choose what sort of terminal Red Ryder should emulate in its operation. Unless you've been told otherwise or know better, I recommend you choose the TTY choice for an unknown system.

- Serial port: The Modem connected to: option tells Red Ryder which serial port (the modem port or the printer port) you have the modem connected to. Red Ryder can use either port for modem communications, but Apple warns that at higher baud rates the modem port is safer.
- Full-screen clears: Each type of terminal emulation has its own method of clearing all or part of the last 24 lines of text received. Clearing all 24 lines is known as a full-screen clear. Normally, a full-screen clear erases the cleared text into oblivion, but by checkmarking the Scroll up lines before full-screen clear option, you can have Red Ryder "push" the lines to be cleared up into the Saved Screens Buffer instead of just erasing them. If you are using TTY emulation, the Ignore TTY full-screen clears option must not be checkmarked in order for this option to work.
- RLE Graphics: CompuServe (a large commercial host system), has the ability to send certain kinds of information (like radar weather maps or pictures of the FBI's Ten Most Wanted Fugitives) using either medium or high resolution graphics. Red Ryder supports both of these graphics modes (both are referred to as 'RLE' or 'Vidtex' graphics by CompuServe) if this choice is selected. If when you tell CompuServe to send a graphic item it displays a message to the effect that your terminal is not capable of receiving the graphics, just ignore that message and proceed. The graphics image will be displayed in a window of its own, and will be framed by a rectangle when the complete image has been received. At this point of Red Ryder's evolution, there isn't much you can do except admire the pretty picture. However, you can use the standard methods (pressing COMMAND-SHIFT-3 or COMMAND-SHIFT-4) to send the window display to a MacPaint compatible disk file or to the printer. This option should be turned off when you are not communicating with CompuServe. Because of a control code conflict, it cannot be used with VT52 emulation.

Terminal Emulation Specifics

Red Ryder can emulate three terminal types: a Teletype (TTY), a DEC VT100, and a DEC VT52. The type of terminal Red Ryder emulates is selected by the **Terminal Emulation Preferences** command under the **Customize** menu. Let's look at each of the terminal types and how Red Ryder emulates them.

Terminal emulation defines how Red Ryder reacts to certain control characters sent by the remote system. TTY emulation, for instance, has a few very simple commands for clearing the screen, moving the cursor to the next line, backspacing, tabbing, and sounding a bell. VT52 emulation adds several additional screen formatting controls, such as those to allow the quick placement of the cursor anywhere on the screen. VT100 emulation is the most advanced of the three, allowing such things as selective scrolling, character attributes such as boldface and underlining, and tab stop placement under host system control.

TTY is what you should use the first time you call if you don't know what the remote system expects. If the remote system asks you to explain what terminal type you have and you don't see "TTY" listed, look for other descriptions that imply a non-specific terminal type, such as "CRT" or "OTHER". VT52 and VT100 emulation is provided in Red Ryder mainly for those who are experienced with using the real thing, since no attempt is made to duplicate the VT100 or VT52 User's Manual here. Those who are familiar with these terminals and their use should have little trouble adjusting to the emulation done by Red Ryder.

VT52 Emulation Details

The VT52 is emulated in the same manner a DEC VT100 operates in its "VT52 Compatible Mode". All of the emulation attributes are preserved, including the "Keypad Mode" (which can be set or reset manually by selecting the VT100 Keypad Mode and VT100 Cursor Key Mode choices under the VT100 Modes menu choice) and the special graphics character set.

VT100 Emulation Details

A lot of terminal programs purport to emulate a DEC VT100. I can guarantee that many don't even come close. From a programmer's standpoint, doing a good VT100 is easy. Doing a very good VT100 is difficult. Doing a <u>usable</u> VT100 is so damn hard that it's rare. <u>Every VT100</u> emulation has inherent deficiencies that you should be aware of. Rather than "forgetting to mention them", I think it's better to come right out up front and tell you exactly what you can expect.

As much of the native VT100 environment as practical is emulated in Red Ryder. The functions not emulated will not be of consequence to the vast

majority of users, but are listed here as a matter of integrity. The functions not emulated include:

- 1) Double-high/double-wide and Single-high/double wide characters. Probably will be supported in a future version.
- 3) Screen mode (inverse screen mode not supported).
- 4) Interlace mode (hardware dependent will not be emulated in future!).
- 5) Blink character attribute (shown as inverse attribute).

I recommend selecting Flashing Cursor with the Terminal Emulation Preferences menu command to help you easily locate the position of the cursor when it rests on top of a character having the inverse attribute.

The VT100 terminal has several "modes" of operation that affect what characters the keypad and cursor movement keys send, as well as the appearance and function of the terminal display. Several of the VT100 modes can be manually set or reset through choices through the VT100 Modes command under the Customize menu. A mode is considered "set" when a checkmark appears in its dialog box choice. The choices in the dialog box this command presents are:

Graphics Character Set Available - Red Ryder supports both the VT52 and VT100 special graphics character set as well as the United Kingdom character set. Selecting this choice allows the remote computer to select and use them.

VT100 Wraparound: When set, the cursor jumps to the beginning of the next line when it reachs the rightmost column. When reset, the cursor remains in the rightmost column and characters are overwritten in that column.

VT100 Autorepeat: When set, most keyboard characters repeat continuously when held down. This includes the keys on the VT100 Status Bar. When reset, no keys are repeated.

V1100 Cursor Key and VT100 Keypad: when set or reset, these modes affect the control codes that are sent for their respective keys.

VT100 Newline: When set, a received linefeed character moves the cursor to the leftmost position of the next downward line. The RETURN key sends both a carriage return and linefeed character. When reset, the linefeed moves the cursor downward vertically only, and the RETURN key sends only a carriage return.

Relative origin: this mode affects how cursor movement sequences are handled by the terminal. You normally should never have to touch this mode as the remote application should do it automatically.

Smooth scroll: When set, the screen will scroll in a smoother fashion than

when it is not set. Although it is easier on the eyes, smooth scrolling is also slower than the normal scrolling.

If you are not experienced with the consequences of the above modes, do not manually change them (unless the remote application specifically tells you to) without consulting the system operator of the remote system. The default settings should work with the majority of remote applications. If things work in strange ways upon starting up Red Ryder, the remote application probably didn't reset these modes properly. Try clicking your mouse in the **Reset Terminal** button to cause Red Ryder to return to the settings it had when first executed.

If you forget how the numeric keypad is mapped during a session, just click in the **Show Keypad Map** button for a picture of the mapping. Note that this map is only guaranteed accurate for the original Macintosh 512K and Plus keyboards, not those supplied with the Macintosh SE, II, or later models. Using the **Install Keypad** utility (described in the chapter "Redefining The Standard VT100 Special Keys") may also change the keypad mapping from what is displayed with this button.

You can also tell Red Ryder to respond to a certain control character sent by the remote computer (an ASCII code 5 (CTRL-E) for those who wonder) with a predefined string of characters known to VT100 users as the **Answerback Message**. This string may contain up to 30 characters, and may include control characters (such as a ^M for a carriage return) in a similar manner as they are included in Macro Key strings.

The VT-Mouse™

You'll find that jack-hammering the cursor movement keys can be quite tiring and hard to follow. Since we've got a mouse on our Macintosh, let's use it! If you hold down your Option key when VT52 or VT100 emulation is active, you'll see that the mouse cursor turns into a small rectangle whenever it is moved into the text area of the display screen as shown in the following picture:

people to come to the a to come to the did of the ome to the aid of their

You'll find that this rectangle fits snugly around a character of text as shown above. Clicking your mouse will cause Red Ryder to send the appropriate series of cursor movement keys to move the cursor to that row and column. A great deal of time has been spent to make sure that this routine works as documented, but I've found that some VT100 editors use some very bizarre input throttling and cursor movement optimization methods. The heuristics incorporated into the VT-Mouse feature are as follows:

If the "VT-Mouse waits for host" option is checkmarked:

1) When you execute the VT-Mouse, it looks first at where the cursor currently is and then where it needs to go. It sends the proper cursor movement key codes

to move in the right direction towards the destination.

2) It waits until the host sends the codes back to Red Ryder which mean "I got your request to move. OK, move it!". If this isn't received within 5 seconds, the VT-Mouse routine gives up.

3) It looks at where the cursor is now and repeats this cycle until it gets to where

you originally specified you wanted it to go.

If the "VT-Mouse waits for host" option is not checkmarked:

1) It looks first at where the cursor currently is and then where it needs to go.

2) It sends <u>all</u> of the cursor movement codes needed to move directly to the desired ending location without waiting for the remote machine to respond.

I suggest you checkmark the VT-Mouse waits for host option to start with. If it appears that the VT-Mouse is getting caught up in an endless loop, then and only then turn off this option.

Even with this intelligence, it is still possible to overrun the host's ability to process the incoming cursor movement keys. The symptom is a group of seemingly random "garbage" characters dumped in the middle of your text. The solution is to increase the VT-Mouse delay value in the VT100 Modes dialog box. This delay value ranges from 0 to 60. It specifies in 1/60th of a second increments how long to wait before sending the next cursor movement key.

Overrunning can also be caused in some cases through the keyboard and VT100 Status Bar cursor movement keys. The solution is to slow down the keyboard repeat rate (in the Macintosh Control Panel desk accessory) and to use the VT-Mouse (with its added intelligence) rather than letting the VT100 Status Bar cursor movement key buttons auto-repeat.

The VT-Mouse feature is not guaranteed to work inside of all VT100 applications. Think of it this way: if it works, hooray. If it doesn't, you're no worse off than any other VT100 user in the world.

A Special Note About Fonts For International Users

When using TTY emulation, you can choose any font, and any point size that is installed on your Macintosh's startup disk. When using VT52 or VT100 emulation, Red Ryder will allow you to only use the TTY-VT52-VT100 font, and then only in 9 point or 12 point size. This is for two reasons. First, Red Ryder uses the highest 128 characters of this font to hold the boldfaced versions of the normal character set as well as the VT52/VT100 special graphics characters set. Secondly, VT52 and VT100 emulations require a non-proportionally spaced font in either 9 or 12 point sizes.

It is possible to use a different font than TTY-VT52-VT100 in conjunction with VT52 or VT100 emulation so that international, diacritical, and special characters can be used. The method for doing this is explained in the chapter "Using International Or Special Character Sets").

Using A Printer With Red Ryder

RED FLAG NOTE: If you are printing on cut sheet (rather than fanfold) paper, be sure to see the "Form Feed After X Lines" explanation below! If your printer does not respond at all to Red Ryder, make sure it has been installed properly with the Chooser desk accessory.

Red Ryder can make full use of any printer connected to your Macintosh. Several commands use the printer directly:

- The Echo Incoming Data To Printer choice under the Local menu.
- The **Print TEXT File...** choice under the **Local** menu.
- The Print Selected Text choice under the Edit menu.
- The Send Display Screen To Printer button in the General Status Bar.

The Send Display Screen To Printer button is discussed in detail in the Chapter "The General Status Bar" and the Print Selected Text menu command is discussed in detail in the chapter "The Terminal Display Window". Before we discuss the other two printing commands listed above, let's discuss how to set up the printer so that it will work correctly with Red Ryder.

First, the correct printer resource file must be present in the System Folder of the disk used to start up the Macintosh. For example, the ImageWriter series of printers has its printer resource file named "ImageWriter". You must use the Chooser dosk accessory (it's on the System Tools disk provided by Apple with your computer) to select and install the desired printer.

The Printer Preferences... command under the Customize menu allows you to select various options that control how text is printed. This command presents you with a dialog box that contains the following options:

Time/Date Heading

The option Time/date heading on (every page or first page only) will print a title at the top of the selected page or pages telling you when the page was printed if it is checkmarked.

Form Feed At Conclusion Of Printing

The Do form feed when printing finishes choice will instruct the printer to do a form feed at the conclusion of the current printing operation if checkmarked. This may or may not be necessary (or desirable) depending on what type of printer you are using. If the last page of printing does not seem to be ejected from the printer after printing finishes, checkmark this option.

+ (cise) Local (pulldonn Nemu)

Con Select while moving mouse on culation Edit mem

Red Ryder Documentation prent released text

Page 38

Form Feed After X Lines

The **Do form feed after every** " \underline{x} " lines printed option will instruct the printer to execute a form feed after every " \underline{x} " (where \underline{x} is a number from 1 to 255) lines have been printed if checkmarked. This will be necessary when using cut sheet paper rather than fanfold paper, especially with laser printers. It's also useful with fanfold paper as it prevents printing from taking place on the paper's perforation. It also tells the **Time/date heading on every page** option when the next page starts.

Special Printing Sequences

The Before sending, send to the printer and After printing, send to the printer options allow you to pass special printer commands before and after printing text. Because Red Ryder always prints in draft mode using the printer's draft mode font, you might use these options to tell the printer to use condensed print (if you are receiving 132 characters per line, for instance), or to print everything in boldface (if your ribbon is getting weak). The special commands to do these kinds of things are listed in your printer's user manual.

You can specify control characters in the strings sent by these options just as you do in a Macro Key String, by preceding the character with a caret character. To send an ESC (also known as Escape) character, you would type ^[(caretleft square bracket).

The Print TEXT File Command

When the **Print TEXT File...** command is selected from under the **Local** menu, Red Ryder will prompt you for the file to print with a standard file selection dialog box, and will then print the file.

The Echo Incoming Data To Printer Command

When the **Echo Incoming Data To Printer** command is selected from under the **Local** menu, it becomes checkmarked in the menu and Red Ryder will echo all received characters to the printer after displaying them in the Terminal Display Window. To stop echoing the characters to the printer, select the menu command again so that it is uncheckmarked.

This is a good way to get a hardcopy record of an online session. However, if your printer is not fast enough to keep up with the incoming characters, it can slow down your throughput. When this is true, your session will be no faster than your printer, no matter what baud rate you are using. I don't recommend using a laser printer with this command, because these printers are made to accept a complete document all at once rather than a few characters at a time now and then. As an alternative, capture the desired text to a file using the Capture Incoming Data To TEXT File... command under the File menu (this is discussed in detail in the chapter "Getting It From Here To There And From There To Here"), and then choose Print TEXT File... once the capture has ended to print out the file.

Connecting To Remote Systems

Red Ryder was designed to not make assumptions about what is connected to the serial port, be it a modem, a direct cable connection to another computer, a laboratory instrument, or nothing at all. Because the commands to make modems dial, answer, or disconnect are varied and brand specific, you will not find menu commands in Red Ryder like "Hang Up Modem". Instead, Red Ryder allows you to fit the software to your particular application by using a generic **Dial Or Redial A Number...** menu choice, and allowing you to construct customize routines (using Red Ryder's Procedure Language and Macro Keys, which are explained elsewhere in this manual) for other necessary functions.

The **Dial Or Redial A Number...** choice found under the **Service** menu is one of the few modem specific commands in Red Ryder. To work properly, it must be used in conjunction with a Hayes compatible modem. Although you could type a modem dialing command directly on your keyboard (as explained in the chapter "Serial Port Settings"), this menu command gives you the added versatility of letting Red Ryder continuously redial a phone number until a connection is achieved.

Before selecting the **Dial Or Redial A Number...** menu choice, it's usually wise to type an "AT" modem command (followed by pressing Return) or two to make sure the modem is ready to accept a dialing command (it should respond with "OK" if it is).

Once you have selected the **Dial Or Redial A Number...** menu command, you'll be prompted to enter the modem dialing command. This is typically "ATDP" if you have a pulse phone, or "ATDT" if you have a touch-tone phone, followed by a single space, followed by the phone number you wish to dial. If a touch-tone phone is being used, you might enter this:

ATDT 1 (800) 555-1212

The spaces, parentheses, and hyphen in the phone number is ignored by most Hayes compatible modems.

After entering the modem dialing command, you would click either on the Cancel button if you change your mind and want to get out of this function, the Dial button if you wish to dial the number only once, or the Redial Until Connected button if you wish to continuously redial until a connection is established.

Some countries have federal laws concerning how many times a number can be repeatedly dialed mechanically. In the United States, it's 15 times. The dialog box presented when you select the **Dial Or Redial A Number...** menu choice contains an editable value called the **Redial Limit**. This number

ranges from 0 to 255, and tells Red Ryder the maximum number of times to redial a number before giving up. If you are in a country where redialing is unrestricted, you can enter a zero as this value and Red Ryder will redial forever (if you let it).

Once a redial begins, a new window is presented that displays the current status of the redial. To cancel a redial in progress, click inside the close box of this window. Once a connection is established or the redial limit is reached, Red Ryder will continuously beep at you until you click in the close box. Lengthy redials present a perfect opportunity to catch up on reading, or if you choose, continue to work inside of desk accessories.

Be aware that in order for the redial function to work, you must be using a Hayes compatible modem, and the modem must conclude a dialing function by returning one of the three messages: "CONNECT", "BUSY", or "NO CARRIER". It doesn't matter if the "CONNECT" message also returns a baud rate, as in the message "CONNECT 1200". For many Hayes compatible modems, the following modem command is used to enable the above function:

ATQ1V1X1

If your modem returns the message "ERROR" after typing in the above command, try the command:

ATQ1V1

As with all modem commands, the above should be typed directly on the keyboard, followed by pressing the Return key. If all goes well, the modem should return an "OK" message to your screen.

Finally, <u>please</u> be smart about how you use the redial feature. When I used to operate a BBS in my home, it had the bad habit of crashing every day or two (this was before I wrote Red Ryder Host - it was operated on an IBM-PC I had gathering dust in the back room). It was easy to tell when the BBS crashed, because some jerk would then decide to start redialing my voice line (just to see if there was a BBS connected <u>there</u>, I suppose). Of course, he turned off his modem speaker so as not wake anyone in his home at 3 a.m., and therefore couldn't hear me screaming "Hullo?" (or much worse). The net effect was that my phone would ring twice per minute for several hours on end. One night, I got <u>very</u> angry and answered the phone - twice per minute - for over three hours. I suspect he got the message when his phone bill arrived the next month - I hope he was calling from Boise. The moral of this story is, don't redial a number you absolutely aren't positive is connected to a computer.

Phonebooks

Once you start taking advantage of the thousands of commercial and "free" services available by modem you'll no doubt end up losing or misplacing phone numbers, account numbers, and passwords or notes. Red Ryder contains a phonebook feature that allows you to safely keep all of these entities collected in neat order much as you do in a mechanical desktop phone directory.

If you desire, you can create as many phonebooks as you wish, sorted by whatever criteria you wish. To create a phonebook, choose the **New Phonebook...** choice under the **Service** menu. You will be presented with a standard file definition dialog box asking you what to name the phonebook and where to put it. After you give it a name and click on the **Save** button, you will be presented with a dialog box asking you if you wish to password protect the phonebook. If you do not wish to, simply click on the **OK** button. Otherwise, type in the password you wish to use and then click on the **OK** button.

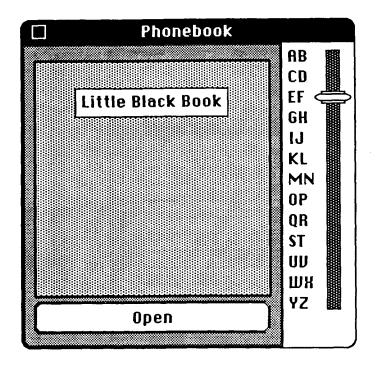
The password can include up to 20 letters, characters, or symbols. It is <u>very</u> important that you type the password slowly and deliberately. If you make a mistake or are unsure, type a backspace and everything you've typed will be deleted, allowing you to start over from scratch.

WARNING! WARNING! WARNING!

Choose a password that you will remember, because password protected phonebooks are encrypted in a very devious manner. This makes them safe from even low-level hacking by unscrupulous sorts and completely useless to those who do not possess the password. If you forget your password, your phonebook and everything in it is lost forever - that's an up front guarantee. The FreeSoft Company will not under any circumstance remove password protection or unencrypt a password protected phonebook for any price or ANY reason, even if you can "prove" that it is yours, Don't bother asking.

For security reasons, a phonebook that is not password protected when it is created can not later have password protection added. In addition, a phonebook that is password protected can not later have that protection removed. Once you make the decision whether or not to password protect a phonebook, you are stuck with it until you or the phonebook dies.

After you create a phonebook with the **New Phonebook...** menu choice, you can open it up by using the **Open Phonebook...** choice under the **Service** menu. A phonebook looks like this when opened:



You can click inside the close box to get rid of a phonebook at any time. The phonebook resembles the mechanical desktop equivalent in appearance and function. The grey box is the "door" under which contains the entries. The name of the phonebook (in this case, I created it as "Little Black Book") is centered in the door. The right edge contains a sliding control knob that lets you open the phonebook to one of thirteen alphabetically sorted "pages" of entries. To open the phonebook to a specific page, simply drag the sliding knob with your mouse until it is next to the letters you desire, then click your mouse in the **Open** button. The door will then open in an animated fashion and the page displayed. To display a different page, drag the control to a different pair of letters. The door will close and the **Open** button will reappear underneath the door.

When the door is "open", the phonebook looks like this:



The Call, Edit, and Delete buttons are inactive until you select an entry by clicking your mouse in it. When selected, an entry is displayed highlighted, as in the entry "Foul Systems, Inc." in the above illustration.

Unlike the mechanical variety, the pages in Red Ryder's phonebooks are endless. When a page contains more than 14 entries, the vertical scroll bar on the right edge of the page becomes active so you can scroll through the various entries to the desired one.

To get rid of an entry, select it and then click on the Delete button.

To add a new entry, click on the **Add** button. A dialog box will appear asking you to supply the following information about the entry:

- Name: This is the name of the entry, that will appear on the selected page when the phonebook door is opened.
- Dialing command: This is the modem dialing command (explained in detail in the chapter "Connecting To Remote Systems") that would be precisely as you would type it into the dialog box presented after selecting the Dial Or Redial A Number... menu choice.
- •Settings file: This is the settings file that you wish to load in before the redialing commences. Use the **Get Settings File** button to easily fill in this entry with a proper file name. This item is optional and can be omitted.
- Procedure file: This is the compiled Procedure File that you wish to execute

after the redialing is terminated. Use the **Get Procedure File** button to easily fill in this entry with a proper file name. This item is optional and can be omitted.

• Notes: This can be anything you want, up to 80 characters of personal notes about the entry, account numbers, passwords, etc. If you are going to put sensitive information in this section, I strongly advise that you password protect the phonebook so that it can not be compromised if stolen or copied. This item is optional and can be omitted.

Once an entry has been added, you can modify the information in it (or view the "Notes" data) by clicking in the **Edit** button.

After selecting an entry and clicking in the **Call** button, the following steps take place in chronological order:

- 1) If the "Settings file" portion of the entry is not empty, the appropriate Settings file is loaded in.
- 2) The dialing command is passed to the **Dial Or Redial A Number** function, and a redial until connection begins as explained in the chapter "Connecting To Remote Systems".
- 3) When the redial terminates (with the YES/NO flag set as explained under the REDIAL command listing in the chapter "Procedure Language Commands"), if the "Procedure file" portion of the entry is not empty, the appropriate Procedure file is immediately executed. If it is empty, Red Ryder will beep at you continuously until you click in the redial window's close box.

Getting It From There To Here Or From Here To There

File transfers are used to achieve several objectives:

- Dumping a text file from your disk over the modem
- Capturing what comes over the modem to a text file on your disk.
- Transferring a file (text or otherwise) that resides on a remote machine to your machine with error checking and correction to insure that it arrives safely.
- Transferring a file (text or otherwise) that resides on your disk to a remote machine with error checking and correction to insure that it arrives safely.

File transfers are usually where beginners get pretty confused and frightened, because they are supported with what seems like a bewildering array of options. Actually, transferring files is not difficult at all in the vast majority of cases, and after a few test runs, you'll be zipping things back and forth like a pro. The options are there only for special circumstances, such as with dealing with a rather "stupid" communications program on the other end, or for transferring files between unlike machines.

In communications jargon, **uploading** a file means sending it from your machine to a remote. **Downloading** means receiving a file on your machine from a remote.

We'll look at the first two objectives listed above, which we refer to as text file transfers. A text file is defined as being composed only of alphanumeric characters. A text file cannot contain non-alphanumeric characters except for tabs and carriage returns (we'll discuss this in more detail later). What it boils down to is that if you choose **Send Text File...** and the file you want to send is not displayed in the file selection dialog box, it isn't a text file and you'll have to use XMODEM or Kermit to send it (and hope the receiver can do something with it once they get it).

Sending Text Files

Under the **File** menu, the choice **Send Text File...** is used to send a text file from your disk over the modem. This is a "raw" dump of the file to the modem which means that the file is sent through the serial port character by character without any error checking or correction. Therefore, telephone line interference can actually transform characters into garbage, or even delete them. There's no way of knowing whether or not the file arrived in the same condition it was sent (other than by the observation of the receiver), so critical data should not be sent by this method (use XMODEM or Kermit instead) when there is a chance of such interference.

You should also know that this type of transfer does not insure that the receiver actually gets the file on their disk. The receiver must instruct their communications software to capture the incoming data to a text file <u>before</u> you send it, and must close the file immediately after the last characters are sent to avoid capturing any additional (and possibly undesirable) characters in the received file.

When **Send Text File...** is chosen, a standard file selection dialog box will appear on your screen. Only files of type TEXT will be shown, to prevent you from sending anything other than a text file. Once you've chosen a file, it will be opened and immediately sent, character by character. During the send, the **Send Text File...** choice becomes **Cancel File Send**, which allows you to cancel a send in progress. Be aware that your keyboard is not locked out during a file send. Any characters you type when the terminal display window is frontmost will <u>also</u> be sent.

During a text file send, you may or may not get any feedback as to the progress of the transfer, depending on the Duplex serial port setting. If you are using Half, Echo, or Null duplex, the characters will be shown in the terminal display window as they are sent. If you are using Full duplex, the characters will not be displayed unless the remote machine echoes the received characters back.

Text File Send Options

There are four Red Ryder options that control how a text file is sent:

- End-of-line character(s). Under the Customize menu, the Keyboard Mapping Preferences... choice brings up a dialog box that contains the option Return Key Sends. If the carriage return and linefeed radio button is selected, Red Ryder will send a linefeed character after every carriage return character it sends from the file. Otherwise, the file will be sent exactly as it is on your disk.
- · Line-by-line prompting. Under the Customize menu, the Text File

Transfer Preferences... choice brings up a dialog with the option Wait after each line sent for <u>x</u> character. If this option is checkmarked, after each line (terminated with carriage return) is sent, Red Ryder will patiently wait for the remote machine to send the character "<u>x</u>" (which you fill in the dialog box with your choice) before sending the next line of text from the file. The prompting character must be alphanumeric - no control characters are allowed for this.

Here's an example of when you might use this option. Let's say that you've typed an electronic mail message into your word processor and saved it as a text file. Now you use Red Ryder to connect and log into your favorite electronic mail service, and wish to upload the message. This saves connect time (and perhaps long distance charges) because Red Ryder can send the file much faster than you can type it while online. For the sake of our example, let's say that the electronic mail service's editor works like this:

```
1> Dear Bob:
2> This is a short test message.
3> _
```

In this example, the editor prompts me for each line by typing the current line number followed by a greater-than symbol (>). Knowing this, I could get into the service's editor, wait for it to prompt me for the first line of entry (it would send a "1>") and then start sending my file. Of course, I would be clever enough to first tell Red Ryder to wait after each line sent for a > character.

• Line-by-line or character-by-character pacing. Sometimes when we are forced to connect to archaic remote machines, we find that they are unable to keep up with the speed which we send them information. The typical result is lost characters. Red Ryder is equipped to automatically use a handshaking method called XON/XOFF which you'll find most up-to-date equipment will deal with in a nice way with no loss of information. If you do discover lost characters, however, don't despair. Red Ryder has two other methods for slowing things down for the remote. Under the Customize menu, the Text File Transfer Preferences... choice brings up a dialog box containing the choice Delay after each. When this option is checkmarked, Red Ryder will introduce delays at certain points during the transfer, depending on how the rest of this option is filled out.

You can select **Delay after each line sent for** \underline{x} **seconds**, where " \underline{x} " is a value from 1 to 9 seconds, and Red Ryder will wait that number of seconds after each line (terminated with a carriage return) is sent before sending the next line of text from the file. Or, you can select **Delay after each character sent for** \underline{x} **60ths**, where " \underline{x} " is a value from 1 to 60, and Red Ryder will wait that number of 60ths of a second after each character is sent before sending the next character in the file.

How do you decide which method (line-by-line or character-by-character) of pacing to use? If in the received file (on the other machine) you discover that characters are lost only at the beginning of lines, you should pace on a line-by-

line basis. If characters are dropping out at various places in the lines, you should use pace character-by-character. To anticipate your next question, I really can't be much help on what values to use for the delays. If a system is quirky enough to require using this option, only experimentation (and perhaps even a reduction in baud rate if possible) will yield the proper values to use.

· Word-wrapping. Up until now, we really haven't discussed too much what a text file is, other than that it's just a file containing text. The way that each line in terminated, however, is very important because some systems must have their cake and eat it too, or they'll just refuse to work with you. Most systems you will encounter will not allow an unlimited line length. Send them a line of text longer than they can handle (usually 80 or 132 characters, but we've seen all sorts of bizarre limits), and they'll choke and puke and make life miserable for you. Therefore, we sometimes will need to make sure that our lines of text do not exceed that limit. Under the Customize menu, the Text File Transfer Preferences... choice brings up a dialog box with the option Wordwrap line ends in sent files at \underline{x} columns, where " \underline{x} " is a value up to 132. If this choice is checkmarked, Red Ryder will load each line of text from the file into memory before sending it. If the line is longer than the limit, Red Ryder will send that line up to the end of the word underneath that limit, a carriage return, and then the rest of the line. The result is a nice paragraph reformatted to the desired column number without any words broken in the middle.

Helpful Hints For Sending Text Files

The difference between a text editor and a word processor is that a text editor works with nothing but characters, but a word processor is capable of including graphics, superscripts, subscripts, and character attributes such as fonts, character point sizes, styles, etc. As you might guess, this type of information is not what we would call "plain text". In order to use the **Send Text File** choice, we must have a file on our disk in plain text format, but there are some things you need to be aware of to accomplish this.

- 1) Choosing Save... from your word processor's File menu will probably save the file in a binary (non-text) format which only that word processor can recognize. If you will be transferring a file to a Macintosh equipped with the same word processor (or one that is able to read your word processor's documents), you should use Send File XMODEM... rather than Send Text File to transfer the file. In order to get a document down to a "plain text" level that can be sent with the Send Text File command, most word processors give you this option when you choose Save As... from their File menus.
- 2) The end-of-line character(s) can be very important for a successful transfer. When the Macintosh was created, a design decision was made that makes Macintosh text files not very compatible with non-Macintosh machines. Most non-Macintosh machines (including IBM-PC's, minicomputers and mainframes) we've worked need two special characters to designate that end of a line, a carriage return character and a linefeed character. The Macintosh only needs a carriage return character, and in fact will deal rather poorly with linefeed characters, displaying them as rectangles at the beginning of each line of text. If

the target machines seems to choke on a file you send it, and you know that the file is plain text, check first to see that it is formatted to the correct right margin column (see Wordwrapping above), and that you are concluding each line with the correct end-of-line characters (see End-Of-Line Characters above).

Receiving Text Files

When the Capture Incoming Data To Text File... choice is selected from under the File menu, you are presented with a standard file definition dialog box. In this box, you type in the name of the file you wish to save information to, and specify where to save the file. Once the Save button is clicked, you will notice that under the File menu, the Capture Incoming Data To Text File... choice has changed into End File Capture. Everything that comes in over the modem will be saved to the file until you choose End File Capture or quit Red Ryder.

What this means is that although you are creating and receiving to a file, what you are receiving does not actually have to come from a file on the remote machine. Since anything and everything received is captured to the disk file, you could be recording an entire session (or parts of a session) for later perusal and editing. Just like with sending text files, no error checking or correction is done. "Garbage" characters caused by phone line interference will be captured along with everything else, so this is not a good way to capture information of a critical nature.

If I wanted to get a disk file containing all of the electronic mail messages that have been sent to me, I could either:

- A) Choose Capture Incoming Data To Text File..., log onto the system, read my mail, log off the system, and then choose End File Capture. Or...
- B) Log onto the system, navigate through the system to the menu that allows me to read my mail, choose Capture Incoming Data To Text File..., read my mail, choose End File Capture, and then log off the system.

If I use the first method, the entire session will be captured in the disk file, including the stuff I really didn't want. The second method is much more selective in that I'm only doing the capture while the information I'm really after is coming over the modem.

There are three options that affect how a text file is received:

• Received Text File Creator. Every Macintosh file is given two signatures when it is created, the file type, and the file creator, both of which are four alphanumeric characters. For text files, the file type will always be TEXT. The file creator tells the Finder what application to start up when you open a document on the Desktop. In other words, if you double-click on a MacWrite document, the Finder looks at the document, sees that it has the same file creator as MacWrite, and therefore knows that it needs to start up MacWrite and pass along the name of that document to open. Since there's no way for Red Ryder to know what you use as your favorite word processor, it gives you

the ability to define what file creator to assign when it creates text files.

Under the Customize menu, the Text File Transfer Preferences choice brings up a dialog box containing the option Received TEXT file creator. You can type in the four characters you wish to assign any text file created by Red Ryder's Capture Incoming Data To Text File... command. Although the following list is not by far complete, it lists a few of the most commonly used text editor/word processor applications' file creators.

Application File Creator
MacWrite MACA
Microsoft Word WORD
Apple MDS Edit EDIT

If you use an application other than the ones listed here, a quick call to the manufacturer's technical support people should yield the proper **file creator** to use. Upper and lowercase letters <u>are</u> different in file creators.

• Stripping Control Characters. Sometimes you'll find that the remote machine is sending special characters that the Macintosh does not need or recognize. This is indicated by "garbage" or rectangle characters displayed when you open up the received file with a text editor or word processor. For instance, if the remote machine sends a linefeed character after every carriage return, the linefeeds will show up a rectangle characters at the beginning of every line. Why? Because the Macintosh believes that a carriage return character is enough to designate the end of a line, and the following linefeed is perceived to be the first character in the next line. Since there is no alphanumeric character equivalent for a linefeed in any of the Macintosh's standard fonts, it is displayed as the "unknown character" (a rectangle).

Under the Customize menu, the Text File Transfer Preferences... choice brings up a dialog box containing the option Strip control characters from received files. If this option is checkmarked, Red Ryder will examine each character received before saving it to the file. If it is anything other than an alphanumeric character or a carriage return, it will discard the character and not save it to the file.

The Strip control characters from received files option can be modified in one other way. Under the Customize menu, the Personal Preferences... choice contains the option Strip control characters allows tabs to pass through. If this is checkmarked, tab characters will also be captured, along with carriage returns and alphanumeric characters. Be aware that the tabs may have been set in different locations on the machine the document was created on than Red Ryder or your text editor/word processor and may need to be reset to their original positions for columns to line up properly.

Sending And Receiving Files Using Error Correcting Protocols

Performing file transfers in a "raw" sense of just dumping characters to the serial port or capturing whatever comes over the serial port is at best risky. Because of telephone line interference, characters can be changed or deleted, and "garbage" characters can be introduced. For this reason, XMODEM and Kermit were created to insure that data arrives in the same condition it was sent. XMODEM and Kermit are similar in that they are both error checking and correcting **protocols**. A protocol is simply a collection of rules for breaking a file up into blocks of data, sending each block, and doing comparisons on the block at both ends. The sending portion of the protocol will continue to retransmit a block until both sides agree that it arrived safely. Both Kermit and XMODEM offer a high degree of reliability (over 99%) and should be used whenever the data is important, or when the file is non-textual in content.

Don't be misled into thinking that XMODEM and Kermit are the same or interchangeable. Although they will appear to you to operate similarly during a transfer, they consist of very different transfer methods. Kermit was developed by Columbia University, and is used most often for transfers with mini and mainframe machines. XMODEM was created by Ward Christenson, and is used mainly for microcomputer to microcomputer transfers. If you are given a choice, use XMODEM because it has a better throughput.

If you're not given a choice, you'll have to go with whatever the remote machine offers. In other words, you can't use XMODEM unless the other machine specifically gives you that option. Using unalike or unavailable protocols will result in you seeing a transfer that never begins.

Performing the actual transfer is <u>very</u> easy, so don't become intimidated. Let's take an example of you sending a file called "FOOBAR" to your friend using XMODEM. Let's assume that you have both connected and are ready to start the transfer.

- 1) You type to your buddy, "I'm going to send you the file FOOBAR. Are you ready?"
- 2) Your buddy types back, "Yes, send away!" It's important to wait until the remote machine tells you to begin. If you are sending (or receiving for that matter) to an online service, you <u>must</u> wait until it explicitly tells you to begin. If you start too early, you could get hung up on the first block, and the transfer will never get started. Remember, wait until the remote machine tells you

something similar to "Start your transfer NOW".

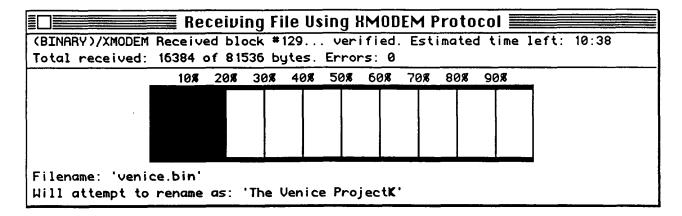
- 3) You go to the **File** menu and choose **Send File XMODEM**. **A standard** file selection dialog box is displayed, and you pick out the file **"FOOBAR"**.
- 4) Meanwhile, your buddy goes up to his **File** menu and chooses **Receive File XMODEM**. He is presented with a standard file definition dialog box, into which he types the file name "FOOBAR".
- 5) Both of you will then see a file transfer status window appear which tells you that the transfer is in progress. After a few moments (don't panic unless it takes more than 30 seconds or so your buddy may be a slow typist or get confused momentarily along the way), the transfer should begin. Everything is automatic from this point on, and you can just sit back and watch until Red Ryder reports the conclusion of the transfer.

In the above example, you could substitute the **Send File - Kermit** and **Receive File - Kermit** menu commands if you desired to use the Kermit protocol rather than XMODEM.

During the transfer, you'll notice that the file transfer status window can be relocated by dragging the mouse cursor in the title bar. You'll also notice that the Apple menu is enabled, which means you can open and work with desk accessories while the transfer is in progress. This gives you the freedom to get something productive done while otherwise you would just be waiting for the transfer to finish. However, if Red Ryder is loaded as an application under Apple's Switcher program, you must not switch out of Red Ryder during the transfer or it will fail. Also, while a transfer is in progress, especially at baud rates higher than 300, don't do things (like hold down a menu for more than a couple of seconds or save a very large document to disk) that completely tie up the Macintosh for an extended period of time. Normal typing takes up very little processing time and shouldn't interfere with the transfer.

Because it's impossible to list everything you can or can't or should or shouldn't do during a transfer, we'll leave it up to you to experiment with various desk accessories during transfers. The worst that could happen is that the transfer will fail, and will have to be restarted.

Let's take a look at the contents of the file transfer status window during a file transfer.



Let's discuss the contents of this window from top to bottom, left to right. The title bar tells us the kind of file transfer being done. The leftmost part of the top line of the status display tells us about the transfer, in this case, we are receiving a MacBinary formatted file (The "(BINARY) message tells us that - a "(TEXT)" message would be displayed if the file was not in MacBinary format). An asterisk would be displayed just before this message if the SUPERCHARGED XMODEM™ option were turned on. After the protocol being used (XMODEM in this case), the current block number is displayed. After a block is received, the error checking will be done and either "verified" or "transmission error" will be displayed. Finally on this line, an estimation for the time remaining in minutes and seconds is displayed. Don't be surprised if the seconds portion jumps erratically during the early stages of the transfer - it will get more accurate and steady as the transfer progresses because mechanical and response delays get averaged out.

The second line of the status display tells us how many bytes have been transferred, and if you are sending a file or receiving a MacBinary format file, the total number of bytes to be transferred are also displayed. Finally, the number of consecutive errors for the current block is displayed. Notice that this is the number of consecutive errors, not cumulative. When an error is corrected, the error count is reset to zero.

If a file is being sent or a MacBinary format is being received, a graphic display of the transfer is shown in the middle of the window. In the above display, we can see that roughly 20% of the transfer is complete. In case you're wondering, this display is intentionally oversized, because I personally like to go to the other side of my room and watch television while a long transfer is running, yet still be able to keep my eye on its progress. Longhairs call this ergonomic engineering. Just think, last week I couldn't spell engineer, now I are one.

At the bottom of the window is the file name you typed when starting the transfer (or "Unknown" if the auto-receive option is turned on and you are receiving a file). If you are receiving a MacBinary format file, the original file name (which Red Ryder will try to rename the file to after it's received if a file by that name doesn't already exist) is also listed.

As an aside, the bizarre looking character at the end of "The Venice" was probably meant to be a trademark symbol, which has been VT100 graphics character in the TTY-VT52-VT100 font.

What Is MacBinary?

Macintosh disk files are very different from files on any other current microcomputer in at least one regard. Most non-Macintosh computers still data in one big clump. The Macintosh, however, actually divides each file up into two subfiles, which are called forks. The data fork is equivalent what non-Macintoshes use to hold the contents of a file. But a Macintosh can contain such things as icons, pictures, dialog boxes, and a whole mess of other things that don't have an equivalent on non-Macintosh machines. For this reason (and others we don't need to get into), the designers of the Macintosh decided to give each disk file a resource fork to hold such things.

Finally, it is desirable to know other things about a file (called **Finder information**) that should be included in a file transfer. Some of these things are pretty esoteric as far as you're concerned, but they do include some things like the original file name, the file's creation and last modification dates, which icon should be displayed by the Finder for that document, and the file type and file creator (these two things were discussed previously in this chapter).

What this boils down to is that when we send a Macintosh file, we need to include three separate entities: the data fork, the resource fork, and the Finder information. The authors of all major Macintosh communications software met and agreed up a standard for doing this. This standard is called **MacBinary**, and guarantees that a file sent by one Macintosh will arrive in the same condition when received by any Macintosh communications program that supports this standard.

The beautiful thing about MacBinary is that it is completely invisible to a non-Macintosh, meaning that you could send a Macintosh application or document to a non-Macintosh, and then that non-Macintosh could then send the file to a second Macintosh. Since the non-Macintosh does not recognize MacBinary, it simply stores the file serially on it's disk as it is received. However, the second Macintosh recognizes that the file is in the MacBinary format as it receives it, and does the conversions necessary to restore the file to its original condition.

Red Ryder will send any file that does not have a file type of TEXT using the MacBinary format. A file with the file type TEXT will only have its data fork sent, not its resource fork or Finder information, so that it can be used by text editors and word processors on non-Macintosh machines.

What Is YMODEM?

YMODEM is nothing more than a repeating XMODEM transfer of a group of files. If you've got fifty small files to send, for instance, it is a pain in the neck to have to keep sitting behind the machine, picking and sending one file at a time.

Red Ryder allows you to create a **batch** of files to send and will then send that batch of files one at a time automatically without further intervention from you necessary. A batch is nothing more than a file containing a list of file names to send.

To create a batch, choose **Create YMODEM Batch...** from under the **File** menu. First, you will be prompted for the name of the batch file you wish to create. This file name will be later selected when you choose the **Send Batch** - **YMODEM** command. You will then be presented with a standard file selection dialog box. Navigate to the file you want to include in the batch and select it. The file selection dialog box will disappear briefly as it adds the file to the batch. The file selection dialog box will then reappear to let you know that it is ready for the next file you wish to add to the batch. Once you've selected all of the files you desire, click on the **Cancel** button to close the batch file.

To send the batch of files, choose **Send Batch - YMODEM** from under the **File** menu. The receiver should choose **Receive Batch - YMODEM** from under their **File** menu. At this point, the first file transfer will begin, with each file in the batch sent in order as the prior file transfer concludes. The receiver is only prompted as to which disk (or HFS directory) the batch of files is to be saved in before the transfer begins.

One important note: if you choose **Send Batch - YMODEM**, the receiver <u>must</u> choose **Receive Batch - YMODEM**, and not **Receive File - XMODEM**. Although XMODEM and YMODEM are closely related in function, they are <u>not</u> interchangeable!

The Auto-Receive Feature

One of the things included in a MacBinary transfer is the file's original name. Many people have expressed surprise that they have to type in a file name before a MacBinary format file is received, only to have the file name they type replaced by the original file name. Wouldn't be easier, they asked, if I just chose **Receive File - XMODEM** and the transfer would begin without further delay, using the file's original name? I relented, and the auto-receive feature was born.

Under the Customize menu, the XMODEM And Kermit Preferences... choice brings up a dialog box that contains the choice Auto-receive using volume/path. When this choice is checkmarked, the auto-receive feature is enabled. Under this choice is the volume (or HFS directory) where files received using XMODEM, YMODEM, or Kermit will be saved if this choice is checkmarked.

Because Red Ryder can not know what disks you'll choose to have mounted when it is run, it always starts out each session with the auto-receive volume/path set to the same volume (or HFS directory) that contains Red Ryder. This volume/path can be changed by choosing **Set Auto-Receive Volume...** from under the **File** menu. The selected volume/path will remain in effect for the remainder of that session (or until it is changed during that session). Just

choosing **Set Auto-Receive Volume...** will <u>not</u> have any effect on file transfers unless the **Auto-receive using volume/path** choice is checkmarked under **XMODEM and Kermit Preferences**.

The auto-receive feature does several things:

- When you choose Receive File XMODEM, Receive File Kermit, or Receive Batch YMODEM, you will not be prompted for a filename or where to save the file. Instead the file transfer will begin immediately.
- If the file is in MacBinary format, the file's original name will be used, unless a file by that name already exists on the disk. If this happens, you will by notified and asked to supply a different file name at the conclusion of the transfer.
- If the file is not in MacBinary format, you will be notified and asked to supply a file name for the file at the conclusion of the transfer.
- If during the session a special auto-receive begin sequence is received, the appropriate kind of file transfer will begin without you having to select a file transfer choice from the File menu. This is a special sequence of control characters that several (but not all) Macintosh communications programs recognize, and it allows the sender to "drive" your machine during a session. Red Ryder will send this sequence before starting any file transfer if the choice Before sending, instruct remote Red Ryder to auto-receive is checkmarked in the XMODEM and Kermit Preferences... dialog box.

It is possible (although rare) for line noise generated "garbage" characters, or "trojan horse" messages on BBS systems (which are in this case harmless, but annoying) to be interpreted as an auto-receive begin sequence. If you find Red Ryder going into a file receive mode at odd times, simply turn off the auto-receive feature and these sequences will be ignored.

XMODEM And Kermit Preferences

Under the Customize menu, the XMODEM And Kermit Preferences choice presents you with a number of options. For each of these options, you should consider XMODEM options to also affect YMODEM transfers in a similar manner.

• Timeout. The option Timeout after <u>x</u> seconds controls how long Red Ryder will wait during an XMODEM transfer for a response from the remote machine before either resending a block of data or requesting a resend. The value you type in for "x" (between 1 and 255 seconds) controls the longest amount of time Red Ryder will wait. The normal value for timeouts, 5 seconds, may need to be extended for some mainframe services like CompuServe, since unexpected delays may occur during the transfer. I would suggest a value of 10-20 seconds for such services, if you observe that the line quality is clean, but are still seeing an abnormally high number of timeout errors during a file transfer. When transferring to another microcomputer, I strongly suggest you reset this value back to 5 seconds.

- Special Kermit ^Q handshake. This option, when checkmarked, tells Kermit to use a special kind of signalling at the end of each transferred block. This handshaking is necessary only for certain IBM mainframes, and should not be checkmarked unless the system operator of the remote machine explicitly tells you that "Control-Q handshaking" is necessary. If it is selected and the remote machine does not need it, the transfer will not work properly and will ultimately fail.
- CRC Error Checking. When XMODEM was originally introduced, it utilized a simple form of error checking called checksum. Later, a more reliable method called CRC was introduced. When the option Use and recognize CRC error checking is checkmarked, Red Ryder will first attempt to use CRC error checking when sending a file XMODEM. If the remote machine does not understand how to do CRC error checking, Red Ryder will automatically switch to checksum error checking. To save some time at the beginning of the transfer, you should not checkmark this option if you are absolutely certain that the remote machine does not know how to do CRC error checking. Otherwise, let Red Ryder make the decision by leaving this option checkmarked. When receiving a file XMODEM when this option is checkmarked, Red Ryder will automatically adapt to whatever method of error checking the sender is using. If this option is not checkmarked, Red Ryder will force the sender to use the checksum method of error checking.
- MacBinary format. When the option Use and recognize MacBinary format for non-TEXT files is checkmarked, Red Ryder will use the MacBinary format (explained earlier in this chapter) for sending all files using XMODEM or Kermit that have a file type other than TEXT. If this option is not checkmarked, Red Ryder will only allow you to send files with a file type of TEXT, and will send only the data fork of the file. When receiving a file, if this option is checkmarked Red Ryder will recognize a file in the MacBinary format, and will convert it to its original form at the conclusion of the transfer. If it is not checkmarked, it will save all blocks of data in the data fork of the received file, give it a file type of TEXT, and do no conversion at the conclusion of the transfer.

The only time this option should not be checkmarked is when you will be receiving a non-TEXT file from a non-Macintosh machine. Certain non-Macintosh files (such as Lotus 1-2-3.WKS files) have data at the beginning of a file that may fool Red Ryder into thinking that the file is a Macintosh file in the MacBinary format.

• Sending TEXT files using MacBinary. If the Send TEXT files using MacBinary format option is checkmarked, files with the file type TEXT will be sent using the MacBinary format. This is useful in certain circumstances, such as when the Finder information (such as the original file name, creation date, etc.) is important. Some text editors and word processors also store some information in the resource fork of a TEXT file, such as tab stop positions or the font and point size to use when the document is opened. This information is only sent when the file is transferred in the MacBinary format. You should

checkmark this option only when you will be sending the TEXT file to a Macintosh with a MacBinary recognizing communications program. A non-Macintosh machine will not be able to make sense of a MacBinary formatted file once it has received it.

- Text file creator. The Received non-MacBinary format TEXT file creator option is equivalent to the Received Text File Creator option under Text File Transfer Preferences described earlier in this chapter. The selected file creator will be given to non-MacBinary TEXT files received using XMODEM or Kermit.
- Stripping linefeeds. If the Strip linefeeds from received non-MacBinary format TEXT files option is selected, Red Ryder will strip out any linefeed characters in a non-MacBinary formatted text file when receiving the file XMODEM or Kermit. Unlike the Text File Transfer Preferences option Strip control characters from received files, this option only strips out linefeed characters, and not any other non-alphanumeric characters.
- Auto-receive. This option is discussed in detail earlier in this chapter under "The Auto-Receive Feature".
- Instructing remote to auto-receive. This option is discussed in detail earlier in this chapter under "The Auto-Receive Feature".
- CompuServe 'B' Protocol Recognition. Besides XMODEM, YMODEM, and Kermit, Red Ryder can receive (but not send) files using a fourth error checking and correcting protocol called the CompuServe 'B' Protocol. This protocol is to be used only with the CompuServe timesharing network.

You'll notice that there is no "Receive File - CompuServe 'B' Protocol" choice under the **File** menu. This is because this protocol is initiated in a different manner. When you tell CompuServe to use the 'B' protocol to send you a file, it will prompt for the file name on your machine. What it is asking you is to supply the complete pathname to save the file as on your disk. Don't forget to supply the volume (and HFS folders) name as part of the file name as described in the "What's In A Filename?" chapter. Once this file name is supplied, CompuServe will send a special sequence that Red Ryder will recognize when this option is checkmarked, and the transfer will begin automatically and will appear similar in operation to an XMODEM or Kermit file transfer.

Since XMODEM may not be very reliable on CompuServe during peak traffic hours, we do recommend that you use 'B' protocol with that network during busy periods (when there are noticeable delays in network response).

• SUPERCHARGED XMODEM™. When the SUPERCHARGED XMODEM™ receive choice is checkmarked, Red Ryder will use a special method of responding to the sender that has the effect of speeding up a file transfer with certain networks drastically. This method entails short circuiting

the error correcting mechanism, so if an error is detected, the file transfer will be cancelled by Red Ryder. Therefore, it should be used only when you are certain of excellent telephone line quality. This feature only yields beneficial results with certain timesharing networks, of which CompuServe and Delphi are two. With other networks, like GEnie or microcomputer based bulletin board systems, this option will not make the transfer go any faster, and you lose the benefit of error-correction. Therefore, we recommend that this option only be used with CompuServe and Delphi, and then only for short files (to reduce the risk of a rare error cropping after 99% of a long file has been received) when the telephone line quality is excellent.

• 1K blocks. XMODEM transfers normally transmit 128 bytes of data in every block sent. When the Use and recognize 1K blocks option is selected, Red Ryder will attempt to send 1024 bytes per block. If the remote machine doesn't support this feature, Red Ryder will drop back to 128 bytes per block. If it is not checkmarked, Red Ryder will only use 128 byte blocks when sending. When receiving a file using XMODEM, if this option is checkmarked Red Ryder will adapt to whichever size blocks the sender transmits (including mixed size blocks). If it is not checkmarked, Red Ryder will force the sender to use 128 byte blocks.

Why use 1024 byte blocks? It reduces the amount of overhead sent with every block of data, resulting in faster throughput, especially at baud rates greater than 1200. The down side is that if the line quality is not excellant, there is more data that must be retransmitted after every error is detected, which may negate or even overwhelm the benefits of larger blocks.

Special Notes About Kermit

When transferring a non-MacBinary formatted file with the file type TEXT using the Kermit protocol, Red Ryder will add linefeeds after carriage returns (as all other Kermit programs should) to comply with the Kermit protocol standard. Therefore, if you intend to use a TEXT file with a Macintosh text editor or word processor that will be downloaded with Kermit, we recommend that you select the Strip linefeeds from received non-MacBinary format TEXT files so that these linefeeds will be removed during the transfer.

When requesting a file transfer from a remote machine using Kermit, <u>do not</u> use the remote Kermit's "Server Mode". Red Ryder does not support the Kermit server mode, primarily because there are other ways to accomplish anything the server mode can do. Don't use the server mode commands "GET" or "PUT", but instead use the interactive mode commands "SEND" and "RECEIVE" for transferring files.

Special Note About YMODEM

A YMODEM transfer will automatically turn on the **Use and recognize CRC error checking** option under the **XMODEM and Kermit Preferences** menu choice if it is not already selected.

Getting Rid Of Files

Now and then when you need to clean up or make more space, you'll wish to get rid of disk files while running an application. When this happens, you've got one of two choices:

- Quit Red Ryder and return to the Finder. This is time consuming and the least desirable method.
- Use one of the public domain desk accessories that have a file deletion command. While better than quitting Red Ryder, this does take up a valuable desk accessory slot under the Apple menu, and for that reason may not be desirable.

For this reason, we've included the **Delete A File...** choice under the **File** menu. After you choose this, a standard file selection dialog box will appear allowing you to select the file to delete. When you click on the **Select** button, Red Ryder will not delete the file immediately, but will first bring up a second dialog box to allow you to confirm that the file should be deleted, or give you one last chance to change your mind.

Getting Personal

You now recognize that Red Ryder is filled to the gills with feature after feature, option after option. The very thing that may have seemed so overwhelming to you at first glance now represents more power than is available in any other telecommunications program on Earth.

Red Ryder is by no means a static program. It continues to grow and evolve based on the needs and suggestions of its users. Unlike many other software developers, I not only need, but actively solicit my customers to tell me what's right, what's wrong, what's missing, and what can be improved in my work.

Under the Customize menu, the Personal Preferences choice brings up a dialog box that contains several niceties that have been suggested to me. The following options are contained in this dialog box:

- Don't zap high bits (allow international characters): This option opens up a whole can of worms. Don't checkmark it until you have read the chapter "Using International Or Special Character Sets". Yes, I know that this option appears disabled in the dialog box. That chapter will tell you how to make it enabled.
- Screen saving and printing functions ignore blank lines: Functions like Archive All Screens will automatically skip over all blank lines until it finds the first non-blank line. This option, when checkmarked, however, will ignore all blank lines within a block of text being saved or printed. It has no effect on either the Capture Incoming Data To TEXT File or Echo Incoming Data To Printer menu choices.
- "Delete A File" continuously cycles until "Cancel" is pressed:
 Normally, the Delete A File choice under the File menu will allow you to
 delete only one file for each time it is chosen from the menu. When this option
 is checkmarked, the Delete A File function will continuously cycle between
 selection and deletion of files until you click in the Cancel button in the file
 selection dialog box. It's handy in this manner if you typically delete a bunch of
 files at a time rather than just one at a time.
- "Find Text" uses zooming rectangle when window shifts: Normally, the Find Text choice under the Edit menu will use a graphically zooming rectangle to help you find the found text whenever the window shifts. If this option is checkmarked, the zooming rectangle is not used, and the found text is just graphically inverted (in a flashing manner) a few times.
- Macros Status Bar scrolls when changing sets: If you find the scrolling slow or distracting when changing between macro key sets using the up and down arrows in the Macros Status Bar, this option, when checkmarked,

will skip the scrolling and just display the new set immediately.

- Display title screen at program startup: OK, so now you know the name of the program and who wrote it and all that good stuff. This option, when checkmarked, will skip the pretty intro and get right to the guts of what you're trying to do.
- Cancel procedure in progress when new procedure is executed: Normally, when a new procedure is executed by a Macro Key when another procedure is running, an error message is displayed telling the user to cancel the procedure in progress before running the new procedure. If this option is checkmarked, the procedure in progress will be automatically cancelled, and the new procedure will be executed with no such message given.
- "Strip control characters" allows tabs to pass through: Normally, the Strip control characters option in the dialog box presented by the TEXT File Transfer Preferences choice under the Customize menu will strip out tab characters along with all other control characters besides carriage returns. If this option is checkmarked, tab characters will be allowed to pass through to the file (will not be stripped out) when using the Capture Incoming Data To TEXT File command.
- Confirm choice before: For the functions listed below, Red Ryder normally presents you with a confirming dialog that verifies that you really want to do that. Once you gain a little confidence, you may wish to turn off this confirming dialog for any or all of these functions:
 - 1) Deleting a file.
 - 2) The Send Screen To Printer button in the General Status Bar.
 - 3) The Send Screen To Disk button in the General Status Bar.
 - 4) The Close Box in the Terminal Display Window.
 - 5) The Reset Time And Billing Clocks buttons in the General and Macros Status Bars.
- Buffered keyboard right margin column: Some services will allow many lines of characters to be sent all at once. Others allow only a single line or less. This option allows you to specify the maximum number of characters the Buffered Keyboard will hold before the Return key must be pressed. It can be a number as large as 32,000.

Settings Files

Red Ryder's strength is in its flexibility. There are close to a zillion choices in all of the various menu commands and dialog boxes that allow you to customize Red Ryder to your specification. A **setting** is what we call anything in Red Ryder that you can change. Needless to say, if you use more than one service you're going to find yourself spending a lot of time adjusting these settings. Red Ryder has a better way.

When you start up Red Ryder, it looks for a file called "Red's 10.0 Stuff" (or whatever version number you're using) on the same disk (and in the same folder under HFS) as Red Ryder. If it doesn't find a file by that name, it creates a new one and uses default settings, otherwise it uses the settings in that file. When you quit Red Ryder, it saves the current settings in the "Red's 10.0 Stuff" file so that when you run Red Ryder the next time, you come up with the same settings you had when you last used it.

You can, however, create other Settings Files, save them on disk, and load them back into Red Ryder at any time. To create a new Settings File containing the current settings, select the **Save Settings To Disk...** choice under the **Customize** menu. To load in a previously saved Settings File, select the **Load Settings From Disk...** choice under the **Customize** menu.

Host Mode

When you select the **Host Mode** choice under the **Local** menu, your Macintosh becomes a host system that is capable of sending and receiving files upon command by remote callers. All menus are locked out while the Host Mode is active. To return to normal operation, click your mouse button. Red Ryder will graphically invert the Status Bar area (by displaying it as white upon black) when it realizes you've pressed your mouse button. While you are in Host Mode, a message telling you so is displayed in the Status Bar area of the Terminal Display Window.

Before entering the Host Mode, you may wish to change the Duplex parameter on the General Status Bar to ECHO. This causes Red Ryder to echo back all characters received from the remote caller so that they appear properly on the caller's screen if they are using FULL duplex. If your remote caller complains that the lines echoed back from Red Ryder write over the top of each other, you should select the Return key sends carriage return and linefeed option in the dialog box presented by selecting Keyboard Mapping Preferences from under the Local menu before entering Host Mode. It doesn't really hurt to go ahead and turn on the Return key sends carriage return and linefeed choice before your remote callers complain - the worst that can happen if it's turned on is that they will get double-spaced text.

Host Mode <u>does not</u> answer the phone when it rings, your modem must be an auto-answer type and be configured properly to do this. Nor does Red Ryder have a Host Mode command to disconnect the phone when the caller is finished. This must be done manually by the remote caller. While your modem is busy waiting for calls and connecting, Host Mode is simply waiting for certain commands to come across the serial port. Anything but these commands is ignored (no error messages are given). This makes Host Mode a "low profile" host that is difficult for phone vandals to mess with. With the addition of a password protection routine, as discussed in the Procedure Examples chapter, Host Mode is impervious to illegal entry.

Let's say you've left Red Ryder in Host Mode and have left for work, where they still use IBM-PC's (ummm....perhaps you work for IBM). You call home and connect to your Macintosh's modem. The first thing to do is type 3 or 4 carriage returns to clear the command line Red Ryder is scanning.

Now you're ready to enter a command, or disconnect the call when you're finished. After any command you enter is completed, type in a carriage return or two before entering the next command. It's also necessary to type a carriage return at the end of each command, and you can backspace to correct errors in typing.

Host Mode does not match communications parameters with the caller. If your parameters are set to 1200-N-8-1 before entering Host Mode, the remote caller's parameters must be set exactly the same or no commands will be recognized.

Most of the below commands have an additional component called **filename**. The correct format for this arguement is:

volume name:file name

The colon (":") separates the volume name and file name. In other words if a host mode command was in the format:

MELTDOWN filename

and you wanted to "MELTDOWN" (whatever that means - we're using it as an example) a file named "REACTOR" on a disk volume named "NUCLEAR", you would type in:

MELTDOWN NUCLEAR: REACTOR

You don't need the volume name portion if the file is on the same disk as Red Ryder, but I highly recommend you use it to avoid problems. The commands used by Host Mode are fairly simple and straightforward:

- SENDA filename causes Red Ryder to send to the remote caller filename as it would if you chose Send TEXT File from under the File menu.
- SENDX filename causes Red Ryder to send to the remote caller filename using XMODEM protocol.
- **SENDK filename** causes Red Ryder to send to the remote caller **filename** using Kermit protocol.
- **RECA filename** causes Red Ryder to receive from the remote caller **filename** using ASCII protocol.
- **RECX filename** causes Red Ryder to receive from the remote caller **filename** using XMODEM protocol.
- **RECK filename** causes Red Ryder to receive from the remote caller **filename** using Kermit protocol.
- *CLOSE* manually stops a file receive started with RECA. Note that the asterisks at the start and end of *CLOSE* are required!
- DO filename exits the Host Mode and immediately executes the Procedure File filename. Note that you can get back into Host Mode by using the Procedure command HOST.

The SENDA command does not warn you that it is about to send the file, it immediately does so upon receipt of that command. The SENDX and SENDK commands both send out "Ready to send..." messages so you know the command was accepted and you can begin the transfer.

The RECA, RECX, and RECK commands all send out a "Ready to receive..." message when the command has been accepted and Host Mode is ready for you to begin. The *CLOSE* command sends a "Closed..." message when the file has been closed and the RECA ended.

Do not use the **RECA** command to send files to Red Ryder that contain a **Host** Mode command. All kinds of strange things can happen. Instead, use either Kermit or XMODEM to send these files to Red Ryder in Host Mode.

The Host Mode has a special feature that allows you to use it as a electronic mail center. If a RECA, RECX, or RECK command is specified with a filename that already exists, the new file received is appended to the end of the old file. This is so that Host Mode can be used as a message center, and new messages won't overwrite old ones as they are uploaded. For this reason, write-protect any disks that contain files you don't want destroyed intentionally or by accident.

Here's an example: Bill Smith calls in and commands Host Mode to RECA MAIL:SCOTT WATSON. Host Mode responds "Ready to receive..." and Bill types in his message to me. When he's finished, he types in *CLOSE* on a new line and Host Mode responds "Closed...". He logs off and Ed Jones calls in next. Ed follows the exact same procedure. When I call in later, I command Host Mode to SENDA MAIL:SCOTT WATSON and my messages are sent one after another. When I get back to my machine, I delete the file SCOTT WATSON on the volume MAIL since I've already read the messages and don't need to keep them around.

For security reasons, there is not a Host Mode command to list the files available for sending, but you can type in a short file listing those you want remote callers to know about. Just name it something simple like **FileList**, put it on the same disk as Red Ryder, and tell your callers to type the command **SENDA FileList** after they log on.

You may wish to use ECHO duplex in the Host Mode. This echos back every character Red Ryder receives so that it is displayed on the callers screen properly if they call using FULL duplex. You <u>must</u> make sure your modem is set up so as <u>not</u> to echo back any characters it receives in command mode (for Hayes compatible modems, the command ATEO - that's a zero on the end - should accomplish this) before entering Host Mode or Red Ryder can get into a fight to the death echoing characters back to the modem (which it in turn echos back to Red Ryder, which Red Ryder in turn...you get the point). If you find yourself in one of these bizarre loops, just select **Change Serial Port Settings...** from under the **Local** menu and switch to FULL duplex.

As an epilog to this chapter, let me explain that Red Ryder's Host Mode feature is rather "dumb" when it comes to such systems, as its meant to serve only you, rather than a group of people (especially inexperienced users). The FreeSoft Company produces a stand-alone product nearly as large as Red Ryder called Red Ryder Host that provides for such things as:

- Telephone answering and disconnecting with automatic baud rate switching.
- Password protection and multi-level security clearances.
- Multiple file transfer libraries with fully commented directories.
- Private and multiple category public mail sections.
- A built-in Survey Language for polling users or taking orders for products.
- An easy to use interface that is suitable for first time and inexperienced callers.

among <u>many</u> other features. This is not so much a plug as a warning to not use Host Mode's limited capabilities and bare-bones interface for a job more suited to a dedicated application like Red Ryder Host.

<u>Using International Or Special</u> <u>Character Sets</u>

When using 8 Data Bits and No Parity, Red Ryder normally strips off the Parity bit, which makes only 128 of the possible 256 characters in a font displayable. This is fine for most purposes, because the standards provided by ASCII (American Standard Code for Information Interchange) govern only the first 128 characters. For users who wish to use international, diacritical, or special characters (those typed in conjunction with the Option or Shift-Option keys), like 'æ' or 'ö', this "high-bit stripping" may not be desirable, and can be turned off by checkmarking the Don't zap high bits option in the dialog box presented by the Personal Preferences choice under the Customize menu. Please realize that because the high 128 characters have not been formally defined in a standard, what you think you are sending may not be what the receiver sees (or the reverse) unless both of you are using the same font, so be sure to agree with the remote machine on which font you'll be using to exchange these special characters.

If you intend to send these special characters, don't forget to set the **Control** characters are sent with option in the dialog box presented by the **Keyboard Mapping Preferences** choice under the **Customize** menu to the cloverleaf key so that the Option key can be used to type those characters.

WARNING! WARNING! This option should not be checkmarked without forethought or you may find yourself suddenly unable to communicate with some services. CompuServe, for instance, requires that the high bit be stripped, and if it isn't, what you'll receive will look like garbage. It is because of the potentially hazardous side effect (and the tendency for some users (like me) to checkmark things before reading the documentation about them), that the Don't zap high bits option in the Personal Preferences dialog box is normally disabled and therefore unavailable to the casual user.

Last warning - did you read the last paragraph carefully? To get the Don't zap high bits option enabled so that you can checkmark it, you must hold down the Option key while selecting Personal Preferences from under the Customize menu.

As explained at the end of the chapter "Terminal Emulation Specifics", Red Ryder can use any font installed in your Macintosh in any point size when using TTY emulation. However, if you need to use international, diacritical, or special characters (those typed in conjunction with the Option or Shift-Option keys) with VT52 or VT100 emulation, some minor surgery needs to be done on Red Ryder. It will not require rubber gloves, just a simple short Procedure File. Therefore, if you're not familiar how to create, compile, and execute a Procedure File, please

refer to the chapters covering this before proceding.

We will use the **PUTPARAM** Procedure command to change 3 bytes in the Settings File. Byte number 142, when non-zero, tells Red Ryder that an additional font is to be listed under the **Font** menu when the **Set Font...** choice is selected under the **Local** menu when VT100 or VT52 emulation is being used. The next two bytes (143 and 144) contain the high and low values of the font number to display in the menu. You can get the proper font number for these values by selecting the font while using TTY emulation - the font number is displayed at the top of the point size selection dialog box.

To get the high and low values to put into bytes 143 and 144, take the font number and divide it by 256. This is the high value. Next, take the high value, multiply it by 256, and subtract that number from the font number. The remainder is the low value. Unless you're a programmer, this is all more than likely gobbledegook (which I dare any spelling checker to recognize), which is alright, because unless you're a hacker-type into creating your own fonts, I would highly suggest you use the Monaco font, which is monospaced, comes in 9 and 12 point sizes, and contains all of the international, diacritical, and special characters you're likely to ever need.

To install the Monaco font, the following Procedure is compiled and executed:

```
(Tell Red Ryder we've got an additional font for VT52/VT100)
PUTPARAM 142,1
(Monaco font is number 4, so the high value is zero)
PUTPARAM 143, 0
(The low value is 4)
PUTPARAM 144,4
```

If you decide later on to get rid of this, you can compile the following Procedure:

```
(Tell Red Ryder not to use any additional VT52/VT100 font) PUTPARAM 142,0
```

When an additional font is installed, Red Ryder will use that font for all characters except:

- 1) Boldfaced ASCII characters
- 2) Special VT52/VT100 Graphics or U.K. character set characters in which case it will use the correct characters contained in the TTY-VT52-VT100 font.

Procedure Files

A Procedure file is simply a set of written instructions you want Red Ryder to perform. Sound like a computer program? It is exactly that, and Red Ryder is equipped with a powerful, yet easy to learn and use, computer language of its own with over a hundred different commands. Red Ryder's Procedure language was originally intended to be used for automatic dialing and log on for remote services. It quickly became evident that the possibilities are literally endless, and the Procedure language has grown in quantum leaps and bounds.

Earlier versions of Red Ryder's Procedure language, as well as all other current telecommunications programs on the market use what computer hackers refer to as an interpreter to execute the instructions. This version of Red Ryder sports the next technical leap up: a true two-pass compiler. By compiling the written commands (called "source code") into a very efficient compiled form (called "object code"), Red Ryder offers the following advantages over interpreted commands:

- Extremely fast execution speed, as the translation of the commands to something meaningful to Red Ryder has already been done by the compiler and need not be done redundantly each time the commands are executed.
- Source code privacy. The source code commands are translated to a non-human readable format. In addition, a very devious method of encryption may be performed on the object code at compile time to prevent prying eyes from hacking the object code to find passwords or other critical text strings. With a few lines of code to add password protection (as shown in Example #12 in the chapter "Procedure File Examples"), you can insure that a compiled Procedure File will be absolutely useless to unauthorized persons.

Red Ryder's Procedure Language is something many beginners get intimidated by because of its obvious power and depth, and put off learning to a later date. Please don't confuse this power with complexity. A Procedure file can be as simple as a two-liner that sets the communications parameters and dials a phone number, or a wild bugger that automates an entire session while you sleep. It's up to you how complex you want to get. Start simple, experiment, learn, and have fun! With the examples provided, there's no reason you can't be programming like a pro after just a few hours of leisurely study.

Just like learning any other computer language, the best way to learn Procedure commands is to browse over the following command descriptions (not trying to memorize them), and then look at some examples, referring back to the command descriptions for a more complete understanding. Learn a few commands at a time, biting off more as you feel comfortable. The examples

start easy, and work their way to complex. The beauty of it is that you can make your procedures as simple or complex as suits your needs and ingenuity, so don't overdo it by moving too fast.

You can execute a Procedure file in a number of ways:

- By choosing **Initiate Procedure...** from under the **Service** menu and selecting the compiled Procedure File to execute.
- By double-clicking the mouse button (or single-clicking and choosing Open from the Finder's File menu) on a compiled Procedure File icon in the Finder. If you start up Red Ryder this way, the Procedure file will be immediately executed.
- By naming a compiled Procedure File with the special name "RRJ\$" (without the quotes) on the same disk (or in the same HFS folder) as Red Ryder Red Ryder will execute this file immediately upon starting up. If you create one of these, don't start up Red Ryder by double-clicking a Procedure file icon as only the RRJ\$ Procedure will be executed. The RRJ\$ file can be used to bypass the title screen, or to set up a default configuration you like instead of using one left over from the last session contained in the "Red's 10.x Stuff" Settings File.
- By using the Host Mode command **DO**, which is explained in detail in the chapter "Host Mode". There is also a Procedure File command **DO** for executing one Procedure File by another.
- Through a macro key. The first character of the macro key should be a backslash ("\") followed by the Procedure filename. Don't get the backslash confused with the slash ("/"). Installing a Procedure File filename into a macro key is explained in the chapter "The Macros Status Bar".
- Through a Phonebook entry, which is explained in the chapter "Phonebooks".

During the execution of a Procedure File, the title bar of the Terminal Display Window will show the filename of the Procedure File executing. If at any time you wish to cancel a Procedure File in progress, you will find that the **Initiate Procedure...** choice under the **Service** menu has changed into **Cancel Procedure**. Choose **Cancel Procedure**, and execution will end immediately.

If the Monitor Procedure choice is checkmarked under the Service menu, Procedure commands are shown in the Status Bar area as they are executed to let you see what is happening. Because of the nature of compiled files, some of the commands (like the JUMPTO and GOSUB commands) that are shown may differ from the way they appear in the source code. The Monitor Procedure command may be overridden with the QUIET and LOUD commands during the Procedure's execution, and it will always be automatically uncheckmarked and disabled in the Service menu (for security reasons) whenever an encrypted Procedure File is executed.

Creating A Procedure File

The first step in creating a Procedure File is to type in the list of Procedure commands to execute into a text file. That file is then compiled into object code, and the object code may be executed. A few words about the source code file. A Procedure source code file is created with your favorite editor/word processor. Each line must be 79 characters long or less, and each line must end with a carriage return. The gymnastics involved in coercing some Macintosh word processing applications to do this can be frightening.

If you use MacWrite, make sure you end each line with a carriage return (don't let what you type "wrap around" to the next line, but physically press the Return key at the end of each line). Choose **Save As...** under MacWrite's **File** menu, and click in the radio button marked **Text only** in the file naming dialog box. MacWrite will then ask you if carriage returns should signify line breaks or paragraphs. You should click in the button labeled **Line Breaks**.

Many of us have found out the hard way that most Macintosh word processors are designed to work well with fancy formatted documents with multiple fonts, pictures, etc., but <u>not</u> plain old text files. For this reason, we have included on your Red Ryder master disk an evaluation copy of a desk accessory text editor called "RedWriter". I <u>strongly</u> suggest that you use a desk accessory text editor for creating Procedure source code files. It's always there under the Apple menu, and its beautifully suited for the many minor editing jobs that crop up while using Red Ryder.

Once you've created and saved your Procedure source code file, it's time to compile it into executable object code. To do this, select Compile TEXT File To Procedure... from under the Service menu. You will be prompted to select the text file to compile, and then for the name of the resulting Procedure File. Don't worry about the compile doing anything to your source code file - it only reads it and does not change it in any way. A special window will appear during compilation, and you can click on the Cancel button inside the compiler window to stop compiling the file immediately. If you choose Check Procedure Syntax from under the Service menu, the compiler will perform normally except that it will not create a compiled object code file.

During compilation, the last several lines that have been compiled will be shown in the center of the compiler window. The bottom-most line is the one being currently compiled. Any error messages you may see refer to the bottom-most line displayed.

Because the syntax of Procedure commands is very consistent, there aren't that many different kinds of errors that can happen during compilation (only eight). Most of the time, simple observation of the offending line will disclose a spelling error.

Compiler Errors

Error: Syntax Error (Parameter #XXX)

Probable cause: Missing or unknown parameter, or an illegally

terminated parameter in a list.

Error: Parameter #XXX should be a YYY

Probable cause: Wrong parameter type - it should be the type denoted by YYY in the error message.

Error: Out Of Label Reference Space

Probable cause: Too many label references. To increase the number of allowed label references, choose **Procedure Compiler Preferences** under the **Customize** menu and put a larger number in the **Max.** number of label references item.

Error: Out Of Label Definition Space

Probable cause: Too many label definitions. To increase the number of allowed label definitions, choose **Procedure Compiler Preferences** under the **Customize** menu and put a larger number in the **Max.** number of label definitions item.

Error: No Such Command

Probable cause: Misspelled or illegal Procedure command.

Error: Missing Command

Probable cause: Two part conditional command (such as IF YES or ALERT) lacks second command to execute when condition is true.

Error: No Such Label <XXX>

Probable cause: The label XXX was referenced but never defined.

Error: Line Is Over 79 Characters Long

Probable cause: Command line is too long or is not terminated with a carriage return.

Procedure Command Syntax

A Procedure command has either one or two parts. There will always be the command name, a label definition, or a comment. Many Procedure commands will also have a parameter list of one or more parameters.

Comments

Any line that begins with a left parenthesis is considered to be a comment, not a command. Since comments are not compiled or included in the object code

file, they do not affect execution speed of the Procedure. Therefore, it's wise to make liberal use of comments in your source code file. Don't try to put a comment after a command on the same line, as in:

MBAR 3 (Show the Macros Status Bar set #3) as this would be construed by the compiler as a syntax error. Instead, put the comment directly before or after the line it describes, whichever suits you best. (Show the Macros Status Bar set #3) MBAR 3

By the way, the right parenthesis at the end of a comment is not necessary, but does makes the source code prettier and more readable.

Label Definitions And References

It is often desirable to branch to different locations in the Procedure File depending on what the remote machine does. We use labels to mark the locations that can be branched to in a Procedure File. A label definition (where execution will branch to) begins with a colon, followed by the label name. The label name may be as long as you like (as long is it follows the 79 character Procedure line limit), but only the first 20 characters are significant. The label name can contain any characters, including spaces. Here are two examples of label definitions as they might appear in a Procedure source code file:

(The following line is a label definition)
:FIRST
BELL
JUMPTO SECOND
END
(The following line is a label definition)
:SECOND
GOSUB THIRD
END

In this example, the labels FIRST and SECOND are defined. Two commands, JUMPTO SECOND and GOSUB THIRD, make references to labels. Notice that the colon is <u>not</u> included in a label reference. If you've got a sharp eye, you probably realize that this Procedure would not compile correctly because the label THIRD was referenced but never defined. Labels can be defined but not referenced, but the reverse is a no-no.

Parameter Types

There are several different parameter types used by Procedure commands:

<u>Label</u>

A label, as described above, is abbreviated in Procedure command listings as LABEL.

Numeric Constant

A numeric constant is abbreviated in Procedure command listings as NUM_CON. It is a number from -2,147,483,648 to 2,147,483,647. Although this is a huge range of numbers, a numeric constant must be a whole number, not a decimal or fraction.

Examples:

123

10

-200

0

44

Character Constant

A Character constant is abbreviated in Procedure command listings as CHAR_CON. A character constant is a single capital letter from A to Z. Examples:

Α

В

C

D

Ζ

String Constant

A string constant is abbreviated in Procedure command listings as STR_CON. A string constant is a group of one or more alphanumeric characters. Typed spaces are considered by the compiler to be a valid part of the string constant. Examples:

Н

HE

HELLO THERE

H1234

23ZZH

Numeric Variable

A numeric variable is abbreviated in Procedure command listings as NUM_VAR. A numeric variable is a capital letter from A to Z followed by a percent sign. There are therefore 26 available numeric variables. Numeric variables are used to hold a numeric constant, and therefore have the same range as numeric constants of the size of numbers they can hold. Numeric variables retain their values even after a Procedure stops execution so that they can be used to pass values between different Procedure Files. Examples:

A%

/ \ /\

В%

C%

D%

Z%

String Variable

A string variable is abbreviated in Procedure command listings as STR_VAR. A string variable is a capital letter from A to Z followed by a dollar sign. There are therefore 26 available string variables. String variables are used to hold a string constant or a character constant, and can hold up to 132 characters, which is by no coincidence the size of the longest line Red Ryder can display in its Terminal Display Window. There are also 3 special string variables referred to as @ variables, because they are designated as the @ symbol followed by a 0, 1, or 2. @ variables are used by certain Procedure commands for displaying information to the user in a Special Status Bar. They are just like string variables, but have a limit of 80 characters. String variables retain their values even after a Procedure stops execution so that they can be used to pass values between different Procedure Files.

Examples:

A\$

B\$

C\$

D\$

Z\$

@0

@1

@2

Numeric Expression

A numeric expression is abbreviated in Procedure command listings as NUM_EXP. Anytime you see a numeric expression parameter, it means that you can use your choice of either a numeric constant or a numeric variable for that parameter.

String Expression

A string expression is abbreviated in Procedure command listings as STR_EXP. Anytime you see a string expression parameter, it means that you can use your choice of either a string constant, string variable, @ variable, or character constant.

Numeric Test Operator

A numeric test operator is abbreviated in Procedure command listings as NUM_TEST_OP. There are eight different symbols that can be used where a numeric test operator is specified. They are:

- > (Greater than)
- < (Less than)
- = (Equal to)
- <= (Less than or equal to)
- >= (Greater than or equal to)
- <> (Not equal to)
- & (Binary AND)
- | (Binary OR)

ON Or OFF Expression

An on or off expression is abbreviated in Procedure command listings as ON_OR_OFF_EXP. Where an on or off expression is specified, you should supply either the word ON or the word OFF.

Filetype Expression

A filetype expression is abbreviated in Procedure command listings as FILETYPE_EXP. A filetype expression is used to specify a four character file type as used by the Finder. Therefore, it is four alphanumeric characters. Examples:

AAAA zzzz MACA WORD EDIT

Procedure Command

A Procedure command is abbreviated in Procedure command listings as PROC_CMD. Certain Procedure commands are actually made up of two Procedure commands. If a certain condition exists (depending on the command) the second Procedure command is executed, otherwise it is not executed. For example:

IF YES JUMPTO PROCESS

is actually constructed of two Procedure commands, **IF YES** and **JUMPTO PROCESS**. If the "yes" condition exists (which is explained in more detail later), the **JUMPTO PROCESS** command is immediately executed.

Flags

There are two internal flags used by some Procedure commands. These flags are used to reflect a certain result of an operation, so that the Procedure execution may branch to perform certain instructions based upon what has happened thus far. The first flag is referred to as the YES/NO flag. It is either set to a "YES" condition, or a "NO" condition. The second flag is called the ERROR flag. It is either set to an "ERROR" condition, or a "NO ERROR" condition. How and where these commands are used are described in the next chapter in the descriptions of Procedure commands that affect or use them. Both flags retain their values even after a Procedure stops execution so that they can be used to pass information between different Procedure Files.

Whenever a Procedure File is executed, the Error flag is set to "NO ERROR", and the Yes/No flag is set to "YES".

How Procedure Commands Are Documented

In the next chapter, each Procedure Command is documented as in the following fictitious command example:

MELTDOWN NUM VAR, NUM EXP

Description: This command blows up the Macintosh attached at AppleTalk node number NUM_VAR only if NUM_EXP is a negative number. Otherwise, it merely laughs at you.

Example:

TYPE Thar she blows!

MELTDOWN A%, -23

SEE ALSO: SELFDESTRUCT, CHERNOBYL.

Refer to Example #2 and #23 for sample usage of "MELTDOWN".

The first line gives the syntactic description of the command with the command itself in boldface and the parameter list (if any) in plain face type. The parameters are listed in order, with any delimiting characters shown - in the above example, we know that NUM_VAR and NUM_EXP must be separated with a comma. If two or more parameters are of the same type, the parameters are numbered to differentiate them, as in NUM_VAR1 and NUM_VAR2. If a parameter is optional (meaning it can be left out in the source code and Red Ryder will use a described default value), it is enclosed with square brackets. Next comes a description of what the command does and what each parameter designates. Next, a list of any Procedure commands used in conjunction with the described command is presented for you to cross reference. Finally, a short example usage may be given, and/or any examples of the command usage in the chapter "Procedure File Examples" are cross referenced for you to examine.

The Procedure commands are separated into 9 categories grouped according to primary function, and are for the most part listed alphabetically under each category. Whenever two or more commands are used in conjunction, they are listed together. The categories are:

- Screen And Keyboard Input And Output
- Variable Manipulation
- File Transfers
- Disk File Manipulation
- Printer Manipulation
- Settings Manipulation
- User Defined Menus
- Procedure Execution Modifiers
- User Defined Dialog Box Commands

Final Hints And Tips

Always put a **PAUSE** command after a **PROMPT** command. Many remote systems need a moment or two to settle down before you deluge it with characters, such as through a **TYPE** command.

If you just can't make a PROMPT command work, you're probably doing one of

three things:

- 1) Not prompting for the correct string ("YOU HAVE MAIL", "You Have Mail" and "you have mail" are three <u>different</u> strings because of the upper and lowercase letters).
- 2) Inserted a leading, trailing, or unneeded space in the string.
- 3) Never receive the string you are prompting for.

In 99% of the cases where somebody says "my Procedure won't work", the problem has been traced to a faulty PROMPT command. Please, please check these <u>very</u> carefully before calling for help.

Unless a Procedure is ended by a command (such as **QUIT**, **RUN**, or **DO**), it will stop normally after executing the physically last Procedure command line or when an error occurs. Although not all of the Procedure commands are used in the "Procedure File Examples" chapter, each command is given a short usage example in its listing in the "Procedure Commands".

Remember that a string or numeric expression can be either a constant or a variable, but not a combination of both. If you wanted to type a constant, a variable, and finally a carriage return, the command:

TYPE This is my name: A\$^M

would not work. It would need to be broken up into the commands:

TYPE This is my name:

TYPE AS

TYPE ^M

A number of things about how Red Ryder compiles your source code into Procedure Files can be modified by choosing **Procedure Compiler Preferences** from under the **Customize** menu. They are:

- Compiler encrypts procedure files: If this option is checkmarked, a very devious scheme of encryption is used on the Procedure File, which renders the file impervious to the likes of file editing programs and pseudo-hackers searching for password strings in the file. If you choose encryption, the Monitor Procedure choice under the Service menu is uncheckmarked and disabled when the Procedure File is executed. Only explicit LOUD Procedure commands will allow tracing of an encrypted Procedure File.
- Max. number of label definitions: This value tells Red Ryder how much memory to set aside to keep track of label definitions while compiling. If you get "ERROR: Out of label definition space" errors from the Procedure compiler, you should increase the number in this value.
- Max. number of label references: This value tells Red Ryder how much memory to set aside to keep track of label references while compiling. If you get "ERROR: Out of label reference space" errors from the Procedure compiler, you should increase the number in this value.

• Show compiler window during compilation: Because of the overhead involved in printing each line as it is compiled (as well as its corresponding line number), you can improve the speed of compilation by uncheckmarking this option. When not checkmarked, the compiler window is not shown during compilation, and errors are signalled only by a beep sound. In that event, you could then turn this option back on and then recompile to find out where and what the error was that was signalled.

OK, let's look at the individual Procedure commands. Remember: first time through don't get overwhelmed, just browse. Then, look over the examples, and come back to the descriptions to see how they're being used, or when you have a need for one in a Procedure you're writing.

Procedure Commands

Screen And Keyboard Input And Output

@ ON OR OFF EXP

Description: The "@" command turns on or off, according to ON OR OFF EXP the display of the Special Status Bar. The "@ ON" command does not show the contents of the three @ variables, it simply prepares for the "SHOW@" command by saving internally the contents of the Status Bar before the "@ ON" command was executed. After the "SHOW@" command is executed, the Special Status Bar displays in white text on a black background from top to bottom the three @ variables: @0, @1, and @2. It's useful for displaying pertinent information to the user during the progress of a Procedure, as a dialog box would stop execution and wait for a user response. After the "@ ON" command is executed, the previous Status Bar contents can be restored by ending or cancelling the Procedure, or by executing an "@ OFF" command.

SHOW@

Description: The "SHOW@" command is also used whenever you change the contents of one or more of the three @ variables, and wish to have the Special Status Bar updated with the new messages. In other words, just changing the contents of an @ variable does not automatically change what is shown in the Special Status Bar, an explicit "SHOW@" command must be executed for the new contents to be displayed.

See also: ERASE, ERASE ALL, COPYINTO.

Example: Refer to Example #1 in the chapter "Procedure File Examples" for sample usage of "@ ON", "@ OFF", and "SHOW@".

ALERT1 STR_EXP/PROC_CMD
ALERT3 STR_EXP/PROC_CMD
ALERT3 STR_EXP/PROC_CMD

Description: These three commands let you look for up to three different strings of characters (specified by the STR_EXP parameter) to be received, and if and when they are, the Procedure command PROC_CMD is immediately executed. STR_EXP must be less than 20 characters and upper and lowercase letters are considered different and must match between STR_EXP and what is received. These commands do not hold up execution of the Procedure while waiting for the string to come over the modem like the "PROMPT" and "PROMPT ^" commands. Control characters cannot be embedded in STR_EXP. PROC_CMD is a full and valid Procedure command to be executed immediately when STR_EXP is received. Anytime an "ALERT" or "PROMPT" STR_EXP is received or a "PROMPT ^" CHAR_CON is received, all active "ALERT", "PROMPT", and "PROMPT ^" commands are

disabled. There is a single space between the "ALERT" number (1, 2, or 3) and STR EXP, but no spaces between STR EXP, the slash character, and PROC CMD. Don't confuse the slash character "/", with the backslash character "\".

SEE ALSO: PROMPT, PROMPT ^

Example: Refer to Example #2 in the chapter "Procedure File Examples" for sample usage of "ALERT1".

BELL

Description: Causes the Macintosh to emit a short beep sound. Useful to alert you audibly during various stages of a Procedure execution.

Example:

TYPE One ringy-dingy!^M

CLEAR

Description: Clears all lines of text in the Terminal Display Window.

Example:

COMM 300-N-8-1-FULL **DIAL** ATDT 555-1212 PROMPT CONNECT CLEAR

DIAL STR EXP

REDIAL STR EXP

Description: These commands work just like the Dial Or Redial A Number choice under the Service menu. You can embed control characters in STR EXP just as you would in a Macro Key, but you don't need to put a ^M at the end (the carriage return is supplied automatically). STR EXP must be 70 characters long or less. Many people will point out that this and the "TYPE" command are similar. They are with two exceptions: a number dialed using the "DIAL" or "REDIAL" commands rather than the "TYPE" command can be redialed at a later time by choosing the Dial Or Redial A Number choice under the Service menu and you would have to explicitly place a ^M at the end of STR EXP if you used a "TYPE" command.

Example:

COMM 300-N-8-1-FULL

(Make sure the modem is ready)

TYPE AT^M

TYPE AT^M

TYPE AT'M

LONG BREAK SHORT BREAK

Description: The "LONG BREAK" command is used to send a long (3 1/2 to 4 second) modem break signal, and is functionally equivalent to a "TYPE ^@" command. The "SHORT BREAK" command is used to send a short (233 millisecond) modem break signal, and is functionally equivalent to a "TYPE ^!" command.

GETLINE NUM EXP, STR VAR

Description: The "GETLINE" command is used to read a line of text directly from the Terminal Display Window into a string variable.

NUM_EXP must be a positive number from 1 to 24, and is the line number of the last 24 lines received (the most recently received line being line number 24) you wish to read into the string variable STR_VAR.

Example:

(Read in line #1 to variable A\$)

GETLINE 1,A\$

PROMPT ^CHAR CON

Description: The "PROMPT ^" command works just like the regular "PROMPT" command, except that it is used to look for a single control character to come over the serial port, rather than a string of characters. CHAR_CON is an uppercase letter from A to Z, representing the control character you wish to look for. Execution of the Procedure stops until the control character is received. Once the control character is received, execution of the Procedure continues normally. Example:

(Wait for a carriage return (CTRL-M) before continuing) PROMPT ^M

(Some systems choke if we type something right after they send data,) (so we'll put a short PAUSE here to let 'em settle down)

PAUSE

TYPE And away we go...

PROMPT STR EXP

Description: The "PROMPT" command is used when you want a Procedure to hold up until a certain string of characters (designated in STR_EXP) to

be received before the next command is executed. STR EXP can be up to 20 characters long. Upper and lower case are considered significant, and must match between STR EXP and the string of characters received in order to be recognized as a match. Control characters may not be embedded in STR EXP (use the **PROMPT ^**).

Example:

(Dial a phone number, wait for the CONNECT message from my modem, and) (then type my password and a carriage return)

DIAL ATDT 555-1212

PROMPT CONNECT

(It's always a good idea to put a short delay in before typing) PAUSE

TYPE My-Password^M

TYPE STR EXP

Description: The "TYPE" command is used to simulate typing on your keyboard. In other words, if you want the Procedure to send a certain string of characters, you put that string of characters in STR EXP. STR EXP can be a mixture of regular and control characters, just as is allowed in a Macro Key. Four special character strings can be placed in STR EXP to yield the following functions:

Sequence	Function
^\$	Sends a DEL character (ASCII code 127)
^ <u>!</u>	Sends a short modem break signal
^@	Sends a long modem break signal
^#	Drops the DTR serial port line for 1 second
These sequences don	't have to appear alone, they can be embedded with
- OMD TUD!14	

hin a STR EXP's characters.

Example:

(Type the string HELLO followed by a carriage return (CTRL-M) TYPE HELLO^M

(And send a short modem break signal)

TYPE ^!

WATCH ON OR OFF EXP

Description: The "WATCH" command is used to turn the mouse cursor into a watch cursor, which is the standard Macintosh indicator that a certain function will require the user to wait. Because other parts of Red Ryder can take the watch cursor "out from under you" in changing it back to other things (like the I-beam cursor in the Buffered Keyboard status bar or the normal arrow cursor if the mouse is moved over the text display area of the Terminal Display Window), it is recommended that the "WATCH" command be used only when your Procedure is in a tight loop, as during a locked condition (see the "LOCK" command below) or during a

compilation (see the "COMPILE LOUD" and "COMPILE QUIET" commands below).

Example:

(Turn the cursor into a watch during a compilation)

WATCH ON

COMPILE QUIET MYDISK: MYPROCEDURE

(When we get to the next instruction, we know that the compile has) (finished, so turn the cursor back into an arrow)

WATCH OFF

Variable Manipulation

ADD NUM VAR, NUM EXP

Description: The "ADD" command is used to add the value NUM_EXP to the value held in NUM_VAR. Only NUM_VAR is affected by this command. If NUM_EXP is a numeric variable, that variable is not changed.

Example:

(Add 15 to the numeric variable Z%)

ADD 28,15

AND NUM VAR, NUM EXP

Description: The "AND" command performs a binary AND operation on NUM_VAR and NUM_EXP, and places the result in NUM_VAR. Only NUM_VAR is affected by this command.

Example:

(AND in the value 256 to the numeric variable M%) AND M%,256

CONCAT STR VAR, STR EXP

Description: This command appends (concatentates) the string of characters in STR_EXP to the end of STR_VAR. Only STR_VAR is affected by this command. If the resulting concatenation would make STR_VAR longer than 132 characters, any characters past the 132nd are lopped off into oblivion.

Example:

COPYINTO A\$, HELLO

(A\$ now contains the word "HELLO")

CONCAT A\$, THERE

(Notice the space between the comma and THERE - A\$ now contains) (the string "HELLO THERE")

CONTAINS STR VAR, STR EXP

Description: This command tests to see whether STR_VAR contains the characters in STR_EXP. If it does, it sets the YES/NO flag to "YES", otherwise it sets the YES/NO flag to "NO". It does not change either STR_VAR or STR_EXP. Upper and lowercase letters are considered different.

Example:

(Does the string variable B\$ contain the word HELLO?)

CÓNTAINS B\$, HELLO

(If it does, ring the bell)

IF YES BELL

CONVUP STR VAR

Description: This command converts any lowercase letters in STR_VAR to their uppercase equivalent.

Example:

COPYINTO A\$, Hello There

(The string variable A\$ now contains the string "Hello There". Let's) (convert the lowercase letters to uppercase)

CONVUP A\$

(A\$ now contains the string "HELLO THERE")

COPYINTO STR VAR, STR EXP

Description: This command replaces the current contents of STR_VAR with the contents of STR_EXP. Only STR_VAR is changed by this command. Example:

(Put the string of characters "HELLO" into the string variable A\$) COPYINTO A\$, HELLO

(And copy that string into the string variable B\$) **COPYINTO** B\$, A\$

DIVIDE NUM VAR, NUM EXP

Description: This command divides NUM_VAR by NUM_EXP and places the result in NUM_VAR. Only NUM_VAR is changed by this command. If a division by zero would result, no division is done and the ERROR flag is set to "ERROR". Otherwise, the division is done and the ERROR flag is set to "NO ERROR".

Example:

(Put the value 15 into the numeric variable A%)

LET EQUAL A%,15
(And divide A% by 3)
DIVIDE A%,3
(A% contains the value 5)

ELAPSED NUM VAR

Description: This command calculates the number of seconds that have elapsed since the last "SAVETIME" command was executed, and places that number in NUM VAR.

Example:

(Save the current time internally)

SAVETIME

(Now introduce a pause of 6 seconds)

PAUSE 360

(And calculate the elapsed time)

ELAPSED A%

(The numeric variable A% should now hold 6)

EMPTY STR VAR

Description: This command tests to see if STR_VAR is empty (contains no characters). If it is, it sets the YES/NO flag to "YES". If STR_VAR contains characters, the YES/NO command is set to "NO". STR_VAR is unaffected by this command.

Example:

(Erase all characters from the string variable A\$)

ERASE AS

(Trust me, it is - but let's go ahead and test that it's really empty).

EMPTY A\$

IF YES BELL

IF NO TYPE He lied! He lied!

ERASE STR VAR

ERASE ALL

Description: The command "ERASE ALL" will destroy the contents and leave empty all string variables (including the @ variables). To erase individual string or @ variables, use the "ERASE" command in conjunction with the appropriate STR_VAR parameter.

Example:

(Erase all string variables)

ERASE ALL

(Erase only the M\$ string variable)

ERASE M\$

FILLS STR VAR, NUM EXP, STR EXP

Description: The "FILL\$" command replaces the contents of STR_VAR with NUM_EXP number of the <u>first</u> character in STR_EXP. Only the contents of STR_VAR is changed by the "FILL\$" command.

Example:

(Put the string "HELLO" into the string variable A\$) COPYINTO A\$, HELLO

(Erase the contents of B\$)

ERASE B\$

(Now put 15 of the first character of A\$ into B\$)

FILL\$ B\$,15,A\$

(A\$ now contains the string "HHHHHHHHHHHHHH")

GETGLOBAL NUM VAR, NUM EXP

Description: The "GETGLOBAL" command is used to obtain the value of certain special internal variables not accessable in any other manner. These variables are called Global Variables, and are accessed by placing the desired Global Variable number (from 0 to 11) into NUM_EXP. The value of that Global Variable is then copied into NUM_VAR. The Global Variables and values they return are:

```
Function
Global #
             Current emulation (0 = TTY, 1 = VT100, 2 = VT52).
   0
   1
             Current cursor row (1 - 24).
   2
             Current cursor column (1 - 132).
             Current duplex (0 = full, 1 = half, 2 = echoback, 3 =
   3
             null).
   4
             Number of 60'ths of a second elapsed since Macintosh was
             started up.
   5
             Current month (1 - 12).
   6
             Current day (1 - 31).
             Current year (1904 - ????).
   7
   8
             Current hour (0 - 23).
   9
             Current minute (0 - 59).
   10
             Current second (0 - 59).
             Current day of week (1-7: Sunday = 1).
   11
Example:
(Get the day of the week global)
GETGLOBAL A%,11
(Find out if today is Monday -> Is A% = 2?)
TEST A%=2
(If it is, bang a gong)
IF YES BELL
```

GETCOST NUM VAR

Description: This value gets the number of cents currently displayed in the billing clocks (in the General and Macros Status Bars) and returns that value in NUM VAR.

Example:

(How much have we spent?)

GETCOST A%

(A% now holds the amount of money spent in cents)

INSTR NUM VAR, STR VAR, NUM EXP, STR EXP

Description: This command returns in NUM_VAR the character position (from 1 to the length of STR_VAR) that the string STR_EXP is first found in STR_VAR. The search for STR_EXP begins at postion NUM_EXP in STR_EXP. If the string isn't found, NUM_VAR is set to zero. Upper and lower case are considered different in the search.

Example:

LET EQUAL A\$, THE QUICK BROWN FOX IS EVEN QUICKER TODAY (Find the first position of the string "QUICK" in A\$)

INSTR A%, A\$, 1, QUICK

(A% now equals 5)

(Get past first QUICK by choosing a starting character as 6, and) (find position of the second QUICK string)

INSTR A%,A\$,6,QUICK
(A% now equals 29)

LEFT\$ STR VAR, NUM EXP, STR EXP

Description: This command copies the first (leftmost) NUM_EXP characters in STR_EXP into STR_VAR. Only STR_VAR is affected by this command. Example:

(Copy the string "HELLO" into A\$)

COPYINTO A\$, HELLO

(Get the first 3 characters in A\$ and put them in B\$)

LEFT\$ B\$,3,A\$

(B\$ now contains the string "HEL")

LENGTH NUM VAR, STR EXP

Description: The "LENGTH" command calculates the number of characters in STR_EXP and returns that number in NUM_VAR. Only NUM_VAR is changed by this command.

Example:

(Copy the string "HELLO" into A\$ COPYINTO A\$, HELLO

(Now put the size of A\$ into A\$)

LENGTH A\$,A\$

(A\$ now equals 5)

LET EQUAL NUM VAR, NUM EXP

Description: This command takes the value NUM_EXP and copies it into the numeric variable NUM_VAR. Only NUM_VAR is changed by this command.

Example:

(Put the value 535 into the numeric variable Z%)

LET EQUAL 28,535

LOADVAR STR EXP

Description: This command loads from disk a set of saved variables that were saved with a previous "SAVEVAR" command. The contents of all string and numeric variables are destroyed in favor of the new values. The @ variables, however, are unchanged by this command. STR_EXP is the filename of the file created with the "SAVEVAR" command. WARNING: Do not use the "LOADVAR" command with any other file than one created with the "SAVEVAR" command!

MID\$ STR VAR, NUM EXP1, NUM EXP2, STR EXP

Description: This command copies NUM_EXP2 number of characters from STR_EXP starting at position NUM_EXP1 into STR_VAR. Only STR_VAR is changed by this command.

Example:

(Put the string "HELLO" into A\$)

COPYINTO A\$, HELLO

(Copy the first 2 characters starting at position 3 of A\$ into B\$) MID\$ B\$,3,2,A\$

(B\$ now contains the string "LL")

MULTIPLY NUM VAR, NUM EXP

Description: This command multiplies NUM_VAR by NUM_EXP and places the result in NUM VAR. Only NUM VAR is changed by this command.

Example:

(Put the value 5 into the numeric variable L%)

LET EQUAL L%, 5

(And multiply L% by 3)

MULTIPLY L%, 3

NUMIOSTRING NUM VAR, STR VAR

Description: This command converts the number NUM_VAR to a string of characters and copies that string into STR_VAR. Only STR_VAR is changed by this command.

Example:

(Put the value -53 into the numeric variable M%)

LET EQUAL M%, -53

(Convert M% to a string and put that in the string variable A\$) NUMTOSTRING M%,A\$

(A\$ now contains the string "-53")

OR NUM VAR, NUM EXP

Description: The "OR" command performs a binary OR operation on NUM_VAR and NUM_EXP, and places the result in NUM_VAR. Only NUM_VAR is affected by this command.

Example:

(OR in the value 256 to the numeric variable M%) OR M%, 256

REPLACES STR VAR, NUM EXP1, NUM EXP2, STR EXP

Description: This command replaces NUM_EXP2 number of characters in STR_VAR starting at position NUM_EXP1 with the contents of STR_EXP. Only STR_VAR is changed by this command. If the replacement causes STR_VAR to be longer than 132 characters, all characters past the 132nd in STR_VAR are lopped off into oblivion.

Example:

(Put the string "HELLO" into the string variable A\$)

COPYINTO A\$, HELLO

(Replace 2 characters in A\$ starting at character #3 with)

(the string "FOOBAR")
REPLACE\$ A\$,3,2,FOOBAR

(A\$ now contains the string "HEFOOBARO")

RIGHT\$ STR VAR, NUM EXP, STR EXP

Description: This command copies NUM_EXP number of the last (rightmost) characters in STR_EXP into STR_VAR. Only STR_VAR is changed by this command.

Example:

(Put the string HELLO into the string variable A\$)

COPYINTO A\$, HELLO

(Copy the last 3 characters in A\$ to B\$)

RIGHT\$ B\$,3,A\$

(B\$ now contains the string "LLO")

SAVETIME

Description: This command saves the current time in an internal variable. It is used in conjunction with a later "ELAPSED" command, to calculate how many seconds have elapsed between two points in time. Example:

(Save the current time)

SAVETIME

(Now wait until 6:00:00 AM)

WAIT 06:00:00

(And put the number of seconds that elapsed between the SAVETIME) (command and 6:00:00 AM in the numeric variable A%)

ELAPSED A%

SAVEVAR STR EXP

Description: This command saves the contents of all numeric and string variables (except for the three @ variables) into a disk file designated by the filename contained in STR_EXP. The variables can be later restored via a "LOADVAR" command. This command does not change the current state of any variables.

Example:

(EEK! I've used up all my variables and need one more numeric variable) (for a temporary calculation. Better save off my variables, first)

SAVEVAR MYDISK: THEVARS

(Now I can do some temporary work with whatever variables I want) (without having to worry about destroying anything important)

LET EQUAL A%, 5

(And so forth...)

(Now let's bring back my original variables)

LOADVAR MYDISK: THEVARS

(And get rid of the temporary file)

DELETE MYDISK: THEVARS

STRINGTONUM STR VAR, NUM VAR

Description: This command converts the string of characters in STR_VAR to a numeric equivalent and places that number into NUM_VAR. Only NUM_VAR is changed by this command. If STR_VAR contains characters of a

non-numeric nature, NUM_VAR will be set to zero.

Example:

(Copy the string "12345" into the string variable A\$)

COPYINTO A\$,12345

(Convert A\$ to a number and put that number in the numeric variable A\$) STRINGTONUM A\$,A\$

(A% now equals 12345)

SUBTRACT NUM VAR, NUM EXP

Description: This command subtracts NUM_EXP from NUM_VAR and puts the result in NUM_VAR. Only NUM_VAR is changed by this command.

Example:

(Put the value 15 into the numeric variable A%)

LET EQUAL A%,15

(And subtract 10 from A%)

SUBTRACT A%, 10

(A% now equals 5)

TEST NUM VAR, NUM TEST OP, NUM EXP

Description: This command does an arithmetic test between NUM_VAR and NUM_EXP (the kind of test is designated by NUM_TEST_OP) and sets the YES/NO flag based on the result of the test. If the test is true, the YES/NO flag is set to "YES", otherwise, the YES/NO flag is set to "NO". Neither NUM_VAR nor NUM_EXP is changed by this command.

Example:

(Put the value 15 into A%)

LET EQUAL A%,15

(Is 15 less than 25?)

TEST A%<25

(We know it is, so...)

IF YES BELL

TIMEDATE STR VAR

Description: This command takes the current time and date, converts both into a string of characters, and copies that string into STR_VAR. The string is in the format "MM/DD/YY HH:MM:SS"

Example:

(Get the current time and date into the string variable A\$)

TIMEDATE A\$

(A\$ might now hold something like "06/16/87 23:49:26")

File Transfers

All file transfer commands set the Error flag to either "ERROR" or "NO ERROR" to indicate the failure or success of the operation performed.

CLOSE

Description: This command is used to terminate the file received started by choosing Capture Incoming Data To TEXT File from under the File menu, or by using a "RECA" command. If a "CLOSE" command is executed when a file capture is not happening, a "Bad Procedure Command" will be returned and the Procedure will be halted.

Example: (Capture all incoming data to the file MAIL on the disk MYDISK)

RECA MYDTSK: MATL

(And keep capturing until we receive the string "END OF MAIL")

PROMPT END OF MAIL

(Now close the capture file)

CLOSE

FASTDUMP

Description: "FASTDUMP" is a feature that was created to solve one person's specific and unusual need. It is one of those unique things in Red Ryder that cannot be accessed through any other means than through this Procedure command, yet it is unlike other Procedure commands in that it is not intended to be used unattended and in fact requires a certain degree of experience on the user's part. First, let me explain the fellow's problem and then what I proposed as an attempted solution through the "FASTDUMP" command.

This fellow had to get a couple of dozen large text files from an archaic mainframe onto his Macintosh. The problem was twofold in that he had to use 9600 baud and his mainframe did not recognize or support any flow control handshaking. My challenge was therefore doubly difficult. First of all, Red Ryder simply cannot print 960 characters per second on the screen and still support all of its features. Secondly, even if Red Ryder could, a mechanical disk drive (floppy or hard) presents even further significant overhead.

My solution was this. I would put in a special Procedure command that when executed, would bring up a standard Macintosh file definition dialog box. The user would type in the name of the file they wished to capture the incoming text to, and then Red Ryder would go into a special "Fast Dumping" mode. When in this mode, there would be no visual feedback that anything was happening. An extremely tight loop was

executing that did nothing more than look for characters coming over the serial port, and stuff them into the file as they came. Very tight loop. No feedback, no overhead. When the user clicked the mouse button, the file would be closed and that was that.

That's the essence of the "FASTDUMP" command. If used in conjunction with a RAM disk (to eliminate mechanical delays from disk drives), it was quite simply the fastest possible method to capture incoming data to a disk file. In fact, even the parity bits are not stripped, so you must use NO parity and 8 databits with this function.

When the "FASTDUMP" command is executed (it has no parameters, just compile a source file with that single command in it), the following happens:

- 1) You get the file definition dialog box. You type the filename and click on the **Save** button.
- 2) A bell rings one time. This tells you that the "FASTDUMP" function is ready to receive.
- 3) Everything received is stuffed into the file with no modification as quickly as possible, until...
- 4) When you click the mouse button, the bell rings twice and the file is closed. This, of course, means that the sending machine must give you some indication of when the file has been entirely sent, so you know when to click the mouse button.

You will probably never need the "FASTDUMP" command. If you ever do, you will bless Red Ryder for having it available.

RECA STR EXP

Description: This command is equivalent to choosing **Capture Incoming Data To A TEXT File** from under the **File** menu. To close the file,
the "CLOSE" Procedure command is used. STR_EXP is the filename that
data is captured to.

RECK STR EXP

Description: This command is equivalent to choosing Receive File - Kermit from under the File menu. STR_EXP is the filename that the data is received to.

Example:

(Receive the file JUNK to the disk MYDISK using Kermit protocol) **RECK** MYDISK: JUNK

RECX STR EXP

Description: This command is equivalent to choosing Receive File - XMODEM from under the File menu. STR_EXP is the filename that data is received to.

Example:

(Receive the file JUNK to the disk MYDISK using XMODEM protocol)

RECX MYDISK: JUNK

RECY STR EXP

Description: This command is equivalent to choosing **Receive Batch** - **YMODEM** from under the **File** menu. STR_EXP must be present (as any string of characters you desire) but is ultimately ignored. The current auto-receive volume is used for the location to save the batch of files (whether or not the auto-receive feature is selected - the default is the same volume (or HFS folder) that Red Ryder resides in).

Example:

(Receive a batch of files - there doesn't have to be a disk) (or folder named Junk!)

RECY Junk

SENDA STR EXP

Description: This command is equivalent to choosing **Send TEXT File** from under the **File** menu. STR_EXP is the filename of the file to send.

SENDK STR EXP

Description: This command is equivalent to choosing **Send File** - **Kermit** from under the **File** menu. STR_EXP is the filename of the file to send.

SENDX STR EXP

Description: This command is equivalent to choosing **Send File** - **XMODEM** from under the **File** menu. STR_EXP is the filename of the file to send.

SENDY STR EXP

Description: This command is equivalent to choosing **Send Batch** - **YMODEM** from under the **File** menu. STR_EXP is the filename of a file created with the **Create YMODEM Batch** choice under the **File** menu.

SENDCLIP

Description: This command is equivalent to choosing Paste under the Edit menu. It sends the contents of the Clipboard to the serial port.

Disk File Manipulation

COMPILE LOUD STR_EXP COMPILE QUIET STR EXP

Description: The "COMPILE LOUD" and "COMPILE QUIET" commands are used to compile a Procedure source code file into an executable Procedure File. "COMPILE LOUD" shows the Compiler Window during the compilation, while "COMPILE QUIET" does not. It is suggested that you consider using the "WATCH" command to change the mouse cursor into a watch (and later back to the normal arrow) before doing a "COMPILE QUIET" in order to notify the user of a brief delay. The compiled Procedure File is given the same filename as the source code file, with the suffix ".PROC" appended. Both "COMPILE LOUD" and "COMPILE QUIET" set the error flag to "ERROR" if an error occurs during the compilation, otherwise, the error flag is set to "NO ERROR".

DELETE STR EXP

Description: This command is used to delete a disk file. STR_EXP is the filename of the file to delete. No confirmation of the deletion is done, so please be sure that STR_EXP is a file that you are sure you wish to destroy.

GETALL

GETSELECT FILETYPE EXP

Description: The "GETALL" and "GETSELECT" commands are used to specify what kinds of files the "GETFILE" command will allow the user to select from in the standard file selection dialog box. "GETALL" (the default state Red Ryder starts up with) means that all file types are acceptable. "GETSELECT" allows you to limit the files that may be chosen to those of type FILETYPE_EXP.

Example:

(Allow the user to choose any file on the disk)

GETALL

GETFILE A\$

(Limit him to choosing only files of type "TEXT")

GETFILE STR VAR

Description: This command puts up the standard file selection dialog box for the user to choose a file. The filename is returned in STR VAR. Since the filename must fit in a string variable, the dialog won't let the user choose a file which has a filename of more than 132 characters. If the user clicks on the "Select" button, the YES/NO flag is set to "YES". If the user clicks on the "Cancel" button, the YES/NO flag is set to "NO".

Example:

(We want the user to choose a TEXT file, so...)

GETSELECT TEXT

(Now put up the standard file selection dialog box)

GETFILE A\$

(If he clicks on the "Cancel" button instead of choosing a file, the) (YES/NO flag is set to "NO", and we'll just quit)

IF NO END

(Otherwise, the full filename he chose is now in the A\$ string variable)

PUTFILE STR VAR

Description: This command puts up the standard file definition box, which prompts the user to type a filename to create and specify where to create it at. The file is not actually created with this command, to do that, follow up with a "USEROPENO" command. The full filename is returned in STR VAR. If the user clicks on the "Save" button, the YES/NO flag is set to "YES". If the user clicks on the "Cancel" button, the YES/NO flag is set to "NO".

Example:

(Let's find out what to name a file and where to put it)

PUTFILE A\$

(If he clicked on the "Cancel" button, the YES/NO flag is set) (to "NO", and we'll just quit there)

IF NO END

(Otherwise, the filename to create is contained in A\$)

RENAME STR VAR, STR EXP

Description: This command allows you to rename a disk file to a different name. STR_VAR contains the original filename, and STR_EXP contains the filename you wish that file to be changed to. This command cannot be used to move a file from one disk (or HFS folder) to another

location, but just to change its name at its present location.

SCREENDISK

Description: This command is equivalent to clicking in the "Send Screen To "Archived Screens" File Button" in the General Status Bar.

Example:

(Take a "snapshot" of the Terminal Display Window and put it in the) ("Archived Screens" text file)

SCREENDISK

Disk File Input And Output

There are two user "paths" that may be open at one time for use with the following commands. They are known as path 1 and path 2. When you open a file for reading or writing, you assign it one of the unused paths. When you close a file, that path then becomes available for use with another file. In other words, you can have up to two file open for reading and writing at the same time. All paths are closed at the termination of a Procedure File. For an example of how these commands are used in an actual Procedure File, see Example #3 in the chapter "Procedure File Examples".

USERCLOSE NUM EXP

Description: Closes path number NUM_EXP and makes that path available for a "USEROPENI", "USEROPENO", or "USEROPENA" command. If the path is not currently open, this command does nothing - so use it if you're at a place in your Procedure where you're not sure if a path is in use or not.

USEROPENA NUM EXP, STR EXP

Description: Opens the filename STR EXP using path number NUM EXP for Append. Append means that if the file doesn't exist, it will be created as a new and empty file. If it does exist, any further "USERWRITE" or "USERWRCR" commands will be done at the end of the file, so previous data is not destroyed. If the file can't be created, the Error flag is set to "ERROR", otherwise, it's set to "NO ERROR".

USEROPENI NUM EXP, STR EXP

.

Description: Opens the filename in STR_EXP using path number NUM_EXP for Input. Input means that you'll be using only "USERREAD" commands with

that path, not "USERWRITE" or "USERWRCR" commands. If the file doesn't exist, the Error flag is set to "ERROR", otherwise, it's set to "NO ERROR".

USEROPENO NUM EXP, STR EXP

Description: Opens filename in STR_EXP using path number NUM_EXP for Output. Output means that you'll be using only USERWRITE and USERWRCR commands with that path. If "filename" already exists, it will be destroyed and recreated as an empty file by this command. If the file can't be created, the Error flag is set to "ERROR", otherwise, it's set to "NO ERROR".

USERREAD NUM EXP, STR VAR

Description: The procedure disk Input/Output commands are meant to be used with text files that have lines that are a maximum of 132 characters long each and end with a carriage return. This command reads data from path number NUM EXP (a file opened with the "USEROPENI" command) either up to 132 characters or the first carriage return (the carriage return is discarded). The data is copied into the specified string variable STR VAR. If the read is unsuccessful (most likely because you've reached the end of the file) the Error flag is set to "ERROR", otherwise it's set to "NO ERROR".

USERWRCR NUM EXP

Description: Writes a single carriage return to path number NUM EXP (a file opened with either the "USEROPENA" or "USEROPENO" commands), effectively terminating a line for later input by the "USERREAD" command.

USERWRITE NUM EXP, STR EXP

Description: Writes the data in STR EXP to path number NUM EXP (a file opened with either the "USEROPENA" or "USEROPENO" commands). It does not write a carriage return at the end of that data, so be sure and use "USERWRCR" commands where necessary. If the write was unsuccessful (path hasn't been opened, disk full, etc.), the Error flag is set to "ERROR", otherwise it's set to "NO ERROR".

Printer	Manipulation

ECHO ON OR OFF EXP

Description: This command is equivalent to choosing **Echo Incoming To Printer** under the **Local** menu. ON_OR_OFF_EXP controls whether or not echoing should be turned on or off.

Example:

(Start sending everything that comes over the serial port to the) (printer now)

ECHO ON

SCREENPRINT

Description: This command is equivalent to clicking the "Send Screen To Printer" button in the General Status Bar.

Example:

(Send a "snapshot" of the Terminal Display Window to the printer) SCREENPRINT

WRITE STR EXP

Description: This command is equivalent to choosing **Print TEXT File** under the **Local** menu. The filename of the file to print is supplied in STR EXP.

Settings Manipulation

ANSWERBACK STR EXP

Description: This command changes the VT100 Answerback Message (which is displayed in the dialog box brought up by choosing VT100 Modes under the Customize menu) to STR_EXP. STR_EXP may contain control characters.

Example:

(Change the answerback message to "Hello" followed by a carriage return) ANSWERBACK Hello^M

BBAR

Description: This command is equivalent to choosing **Buffered Keyboard** from under the **Local** menu.

Example:

(Activate the Buffered Keyboard)

RRAR

COMM STR EXP

Description: This command is used to change the serial port settings for baud rate, parity, data bits, stopbits, and duplex. STR_EXP is in the form:

BAUD-PARITY-DATABITS-STOPBITS-DUPLEX

where:

BAUD may equal 300, 450, 1200, 2400, 4800, 9600, 19200, or 57600.

PARITY may equal N, O, E, K, or S

DATABITS may equal 5, 6, 7, or 8

STOPBITS may equal 1, 1.5, or 2

DUPLEX may equal FULL, HALF, ECHO, or NULL

Example:

(Set the serial port to 1200 baud, NO parity, 8 data bits, 1 stopbit,) (and FULL duplex)

COMM 1200-N-8-1-FULL

CONTROL1 ^CHAR CON

CONTROL2 ^CHAR CON

CONTROL3 ^CHAR CON

Description: These three commands modify the **Control Character Buttons** in the **General Status Bar**. The buttons are designated #1, #2, and #3 from left to right. CHAR_CON should be an uppercase letter from A to Z.

Example:

(Set up the three Control Character Buttons to be CTRL-A, CTRL-B,) (and CTRL-C)

CONTROL1 ^A

CONTROL2 ^B

CONTROL3 ^C

CRC ON OR OFF EXP

Description: This command is analogous to the "Use and recognize CRC error checking" option in the dialog box brought up by choosing XMODEM And Kermit Preferences from under the Customize menu. A "CRC ON" command would checkmark that option, and a "CRC OFF" command would uncheckmark it.

Example:

(Turn on CRC error checking)

CRC ON

DELKEY ON OR OFF EXP

Description: This command is analogous to the "Backspace key is DEL key" option in the dialog box brought up by choosing Keyboard Mapping Preferences from under the Customize menu. A "DELKEY ON" command would checkmark that option, and a "DELKEY OFF" command would uncheckmark it.

Example:

(Make the backspace key a DELETE key (sends ASCII code 127))

DELKEY ON

(Make the backspace key normal (sends ASCII code 8))

DELKEY OFF

DISPLAY NUM EXP

Description: This command is analogous to the "Display columns" option in the dialog box brought up by choosing Terminal Emulation Preferences from under the Customize menu. NUM_EXP must be a number between 20 and 132.

Example:

(Give me an 80 column display)

DISPLAY 80

ALT ON OR OFF EXP

Description: This command is analogous to the "` key sends ASCII code: 'X'" option in the dialog box brought up by choosing Keyboard Mapping Preferences from under the Customize menu. A "ALT ON" command would checkmark that option, and a "ALT OFF" command would uncheckmark it. This command does not change the ASCII code number portion of that option.

Example:

(I've got 27 in the ASCII code number portion of that option,) (so turn on the `key so that it works like an ESCAPE key)
ALT ON

FONT NUM EXP1, NUM EXP2

Description: The "FONT" command is used to change the font and/or point size currently used in the Terminal Display Window. NUM EXPl is the font number to use, and NUM EXPl is the point size to use. If the font number or point size isn't available, this command has no affect. How does one find out the font number? It is displayed at the top of the point size selection dialog box when you choose Set Font from under the Local menu, and then choose a font from under the Font menu. For

instance, if I choose the Monaco font, the top of the point size selection dialog box would display the message:
Select the point size for use with the font
"Monaco" (font number 4)
Example:
(Use Monaco (font number 4) font, 9 point size)

GBAR

FONT 4,9

Description: This command is equivalent to choosing **General Status**Bar under the Local menu.

Example:

(Display the General Status Bar)

GBAR

LF ON OR OFF EXP

Description: This command is analogous to the "Return key sends:" option in the dialog box brought up by choosing Keyboard Mapping Preferences from under the Customize menu. A "LF ON" command is equivalent to clicking the "carriage return and linefeed" radio button, and a "LF OFF" command is equivalent to clicking the "carriage return only" radio button.

Example:

(Return key sends a carriage return followed with a linefeed) LF ON

LOADSET STR EXP

_____.

Description: This command is equivalent to choosing Load Settings From Disk from under the Customize menu. STR_EXP is the filename of the Settings File to load.

Example:

(Load the Settings File "BBS Settings" which is on the disk "MYDISK") LOADSET MYDISK:BBS Settings

LOUD

Description: This command is equivalent to checkmarking the Monitor Procedure choice under the Service menu.

Example:

(Start monitoring the Procedure command execution)

Red Ryder Documentation

MACRO STR EXP

Description: This command is equivalent to choosing Load Macro Keys From Disk from under the Customize menu. STR_EXP is the filename of the Macro Key file (created by choosing Save Macro Keys To Disk under the Customize menu) to load.

Example:

(Load in the Macro Key file "BBS Macros" on the disk "MYDISK")

MACRO MYDISK: BBS Macros

MBAR [NUM EXP]

Description: This command is equivalent to choosing Macros Status Bar under the Local menu. NUM_EXP is optional. If it is not supplied, the current active set is displayed. Otherwise, NUM_EXP must be either 1, 2, or 3, which changes the current active set to that number.

(Display the Macros Status Bar, set #2)

MBAR 2

MODEM

Description: This command instructs Red Ryder to communicate through the modem serial port.

Example:

(Use the modem port)

MODEM

NOZAP ON OR OFF EXP

Description: This command is analogous to the "Don't zap high bits (allow international characters)" brought up by choosing Personal Preferences from under the Customize menu. A "NOZAP ON" command would checkmark that option, and a "NOZAP OFF" command would uncheckmark it. BE SURE TO READ THE WARNING ABOUT THAT OPTION IN THE CHAPTER "Using International Or Special Character Sets" BEFORE USING THIS COMMAND!

Example:

(Don't zap the high bits)

NOZAP ON.

PRINTER

Description: This command instructs Red Ryder to communicate through the printer serial port.

Example:

(Use the printer port to communicate through)

PRINTER

QUIET

Description: This command is equivalent to uncheckmarking the **Monitor Procedure** choice under the **Service** menu.

Example:

(Stop monitoring Procedure command execution)

QUIET

REDIAL LIMIT NUM EXP

Description: This command is equivalent to the "Redial Limit: 'X' tries" option in the dialog box presented when you choose Dial Or Redial A Number under the Service menu. NUM_EXP should be a number between 1 and 255, or zero if you wish there to be no limit for redialing tries.

Example:

(Set the redial limit to 15 tries)

REDIAL LIMIT 15

RESET

Description: This command sets the elapsed time clock to 00:00:00 and the billing clock to \$0.00.

Example:

(First dial our number)

DIAL ATDT 555-1212

(Wait for a connection)

PROMPT CONNECT

(Now synchronize the elapsed time and billing clocks)

RESET

SCREEN ON OR OFF EXP

Description: This command is analogous to the **Hide Terminal Window** choice under the **Local** menu. A "SCREEN OFF" command will hide the Terminal Display Window (make it invisible), and a "SCREEN ON" command

will make the Terminal Display Window visible. (Make sure the Terminal Display Window is visible) SCREEN ON

SETCOST NUM EXP

Description: This command is analogous to the Set Billing Cost choice under the Service menu. NUM EXP is the amount to increment the billing clock each minute in 100ths of a cent.

Example:

(Set billing cost to \$5.00/hour)

SETCOST 833

SLOW ON OR OFF EXP

Description: This command is analogous to the "Timeout after 'X' seconds" option in the dialog box brought up by choosing XMODEM And Kermit Preferences under the Customize menu. A "SLOW ON" command sets the timeout value to 20 seconds. A "SLOW OFF" command sets the timeout value to 5 seconds.

Example:

(Set XMODEM timeout to 5 seconds)

SLOW OFF

STRIP ON OR OFF EXP

Description: This command is analogous to the "Strip control characters from received files" option in the dialog box brought up by choosing TEXT File Transfer Preferences from under the Customize menu. A "STRIP ON" command checkmarks this option, and a "STRIP OFF" command uncheckmarks this option.

(Strip control characters from received text files)

STRIP ON

TTY

VT52

VT100

Description: These three commands tell Red Ryder what kind of terminal it is to emulate, a TTY (generic) terminal, a DEC VT52 terminal, or a DEC VT100 terminal.

Example:

(Emulate a 80 column VT52 terminal)
DISPLAY 80
VT52

VBAR

Description: This command is equivalent to choosing **VT100 Status Bar** from under the **Local** menu.

Example:

(Display the VT100 Status Bar)

VBAR

XKSTRIP ON OR OFF EXP

Description: This command is analogous to the "Strip linefeeds from received non-MacBinary format TEXT files" option in the dialog box brought up by choosing XMODEM And Kermit Preferences from under the Customize menu. An "XKSTRIP ON" command checkmarks this option, and an "XKSTRIP OFF" command uncheckmarks this option. Example:

(Strip linefeeds from XMODEM received TEXT files) **XKSTRIP** ON

Procedure Execution Modifiers

ATTACH FILETYPE EXP, STR EXP

Description: This command is used in conjunction with the "RUN" command in order to launch a Macintosh application specifying a document for that application to open once it has begun execution. For instance, if you wanted to leave Red Ryder and run a BASIC program, you would use the "ATTACH" command to specify the name of the BASIC program, and then the "RUN" command to execute the BASIC interpreter application. Or, you could use it to cause a spreadsheet program to execute and automatically open up a desired spreadsheet data file.

This command is provided for use by those with above average Macintosh experience, as it requires that you know the filetype, as well as the file name of the document you wish to attach to the application about to be executed. FILETYPE_EXP is a four character Finder filetype, and STR_EXP holds the file name of the document to be attached. Example:

(I want to start up MacWrite and have it open a MacWrite file) (named "Diary". I happen to know that MacWrite documents have a) (filetype of "MACA". Both MacWrite and the document reside on a disk) (named "MYDISK")

ATTACH MACA, MYDISK: Diary RUN MYDISK: MacWrite

DO STR EXP

Description: This command causes the current Procedure File to stop execution, and the Procedure File contained in STR_EXP to be executed.

(Chain off to the Procedure File "Second Proc" on the disk "MYDISK")

DO MYDISK: Second Proc

CRAWL ON OR OFF EXP

Description: Because Red Ryder uses a compiler, the speed at which a Procedure File executes may make the command tracing feature activated by the Monitor Procedure command under the Service menu (and the "LOUD" Procedure command) useless for keeping track of what's happening at any given moment. A "CRAWL ON" command slows down the execution of the Procedure File to a rate of one command executed every two seconds. A "CRAWL OFF" command makes the Procedure File execute at its normal speed. The delay caused by a "CRAWL ON" command is always cancelled when the Procedure File is terminated, so that future Procedure Files are not affected by it.

Example:

(Yikes! Things are running much to fast for me to see, slow it down!) CRAWL ON

END

Description: The "END" command terminates the execution of a Procedure. While Red Ryder will always terminate the Procedure after it has executed the last physical Procedure command in a Procedure File (whether or not that command is an "END" command), the "END" command is useful to terminate the Procedure File at some other place than at the physical end of the File.

Example:

(If the Yes/No flag is set to "YES", branch to the routine GETMAIL) (Otherwise, end the Procedure File immediately)

IF YES JUMPTO GETMAIL

END

:GETMAIL

(If I get here, collect my waiting mail)

GOSUB LABEL

Description: The "GOSUB" command is used with small collections of commands (called subroutines) that are used by various routines in a Procedure. When this command is executed, an immediate branch to Include is performed, where commands are executed in order until the next "RETURN" command. At that point, execution is branched back to the command following the original "GOSUB" command. Subroutines may "GOSUB" to other subroutines to a level of twenty deep, with subsequent "RETURN" commands returning to the command following the most recently executed "GOSUB" command.

Example:

(This routine will branch execution to the subroutine FIRST)
(FIRST will ring the bell once and branch to the subroutine SECOND)
(SECOND will ring the bell once and return back to FIRST. FIRST then)
(rings the bell once more, and then returns back to the command)
(following the original GOSUB command, which halts execution of the)
(Procedure.)

GOSUB FIRST

END

:FIRST

BELL

GOSUB SECOND

RETURN

: SECOND

BELL

RETURN

TIOCH

HOST

Description: The "HOST" command terminates execution of the Procedure File and puts Red Ryder in "Host Mode", which is equivalent to choosing Cancel Procedure from under the File menu, and then choosing Host Mode from under the Local menu. The Host Mode command "DO" may be used to exit the Host Mode and execute a Procedure File. Example:

(Quit the Procedure File here, and immediately go into Host Mode) ${f HOST}$

IF ERROR PROC_CMD

~~~~~~~~~~~~~

Description: If the Error flag is set to "ERROR", the Procedure Command PROC\_CMD is executed. If the Error flag is set to "NO ERROR", the Procedure Command PROC\_CMD is not executed.

Example:

(If there's an error, ring the bell and quit the Procedure)

IF ERROR GOSUB ERROR ROUTINE

(Otherwise, just end the Procedure here)

END

:ERROR ROUTINE

BELL

RETURN

# IF NO ERROR PROC CMD

Description: If the Error flag is set to "NO ERROR", the Procedure Command PROC CMD is executed. If the Error flag is set to "ERROR", the Procedure Command PROC CMD is not executed.

Example:

(If the Error flag is set to "NO ERROR" end the Procedure, otherwise) (ring the bell once and then end the Procedure)

IF NO ERROR END

BELL

END

IF NO PROC CMD

Description: If the Yes/No flag is set to "NO", the Procedure Command PROC\_CMD is executed. If the Yes/No flag is set to "YES", the Procedure Command PROC CMD is not executed.

Example

(If the numeric variable A% is not less than 15, ring the bell)

**TEST** A%<15

IF NO BELL

IF YES PROC CMD

Description: If the Yes/No flag is set to "YES", the Procedure Command PROC CMD is executed. If the Yes/No flag is set to "NO", the Procedure Command PROC CMD is not executed.

Example

(If the numeric variable A% is less than 15, ring the bell)

**TEST** A%<15

IF YES BELL

JUMPTO LABEL

Description: The "JUMPTO" command is used to branch immediately to a different place in the Procedure File, marked by LABEL.

(Branch immediately to the label LABEL1)

JUMPTO LABEL1

(The following BELL command will never be executed!)

BELL

:LABEL1

(Ring the bell once and end the Procedure File)

BELL

**END** 

LOCK ON OR OFF EXP

Description: The "LOCK" command is included for very special circumstances, and should not be used indiscriminantly. The normal policy of execution is for Red Ryder to perform one Procedure instruction, look for and process any incoming data over the serial port or handle user input, execute the next Procedure command, process incoming data and handle user input, and so forth. A "LOCK ON" command places the Procedure file in a closed loop, where Procedure commands will be executed continously without interruption until a "LOCK OFF" command is executed, which puts Red Ryder back into a normal mode of operation.

Be <u>very</u> careful with this command, because incoming data, menu selections, keyboard presses, mouse activity, etc. are completely locked out. If your Procedure goes into an endless loop, the Macintosh would actually have to be turned off and then back on in order to regain control. It is most useful when you have a small number of commands that must be executed in a burst before any waiting incoming data is collected and processed.

Example:

(Look for the string of characters "CONNECT" to come over the) (serial port)

PROMPT CONNECT

(Now open up the file MYDATA on the disk MYDISK, and then look for the) string "PASSWORD?" to come over the modem. If the string came and went) before the OPEN command was finished, I would miss it completely, so) (let's put the Procedure in a locked condition during the open and use) (an ALERT command to do the looking)

ALERT1 PASSWORD?/JUMPTO GOT IT

LOCK ON

USEROPENI 1, MYDISK: MYDATA

(OK, open the floodgate, if the string was received during the) (USEROPENI command, the active ALERT command will see it)

LOCK OFF

(If we got here, the string was not in the waiting data, so we'll just) (use a normal PROMPT command to look for the string)

**PROMPT** PASSWORD?

(In either case, we ought to get to here when "PASSWORD?" is received)

:GOT IT

(Just ring the bell and end the Procedure here)

BELL

END

# ONPANIC PROC CMD

Description: This command places the Procedure Command PROC\_CMD in a temporary holding space, if the "PANIC" condition occurs while a "PROMPT" or "ALERT" command is active, PROC\_CMD is retrieved and immediately executed. The "PANICAFTER" command is used to describe when the "PANIC" condition exists.

Example:

(If the string "CONNECT" does not come over the serial port in 30) (seconds, set the "PANIC" condition and log off immediately)

PANICAFTER 30

ONPANIC JUMPTO GET OUT

PROMPT CONNECT

(If we get here, everything went fine, we'll just end the Procedure) (File here, though)

END

:GET OUT

(Gadzooks! If we got here the "CONNECT" string didn't arrive, so we) (would follow these comments with a routine to hang up the phone.)

TYPE +++

PROMPT OK

PAUSE

TYPE ATH^M

**END** 

#### PANICAFTER NUM EXP

Description: This command is used in conjunction with the "ONPANIC" command to specify how long to wait for a desired string of characters to arrive before giving up and setting a "PANIC" condition. NUM EXP is the number of seconds to wait once a "PROMPT" or "ALERT" command is executed before executing the Procedure Command specified in the most recent "ONPANIC" command. Whenever a "PROMPT" or "ALERT" command is satisfied by an incoming string, the "ONPANIC" command is cancelled and must be followed by another "ONPANIC" command if desired to work for subsequent "PROMPT" or "ALERT" commands.

Example:

(Wait 30 seconds for "CONNECT", branch to "GET OUT" if it doesn't come)
ONPANIC JUMPTO GET OUT

PANICAFTER 30

PROMPT CONNECT

(Type a CTRL-C, then wait for the string "PASSWORD?")

#### PAUSE

TYPE ^C

(The PANICAFTER command was turned off if I got here, so I need to do) (another one)

#### PANICAFTER 30

#### PROMPT PASSWORD?

(Everything worked fine, so type my password and end the Procedure)

#### PAUSE

TYPE MyPassword^M

END

#### :GET OUT

(Whoops, one of the PROMPT's never arrived, so I'll hang up the phone) (and alert the user with a couple of bells.)

TYPE +++

PROMPT OK

TYPE ATH^M

BELL

BELL

END

# PAUSE [NUM EXP]

Description: This command stops execution of the Procedure File for a specified period of time, and then allows the Procedure File to continue execution as normal. NUM\_EXP is a number of 1/60ths of a second to delay. If you wished to pause the Procedure for one second, you would use a "PAUSE 60" command, for three seconds "PAUSE 360". NUM\_EXP is optional - if it isn't supplied, a value of 60 (one second) is used. Example:

(Wait of the string "CONNECT" to come over the modem)

#### PROMPT CONNECT

(Always pause for a second before typing something to make sure the) (remote machine is ready to handle your input. The following command) (will pause for 1 second)

#### PAUSE

(Now, type in my secret password and a carriage return) TYPE MyPassword^M

#### QUIT

Description: This command causes the Procedure File to immediately terminate execution, and then it will cause Red Ryder to immediately terminate execution (equivalent to choosing **Quit** from under the **File** menu) and return to the Finder.

#### Example:

(I'm all done with Red Ryder, quit back to the Finder)

## RETURN

Description: This command is used in conjunction with the "GOSUB" command to end execution of a subroutine and branch back to the Procedure command following the most recently executed "GOSUB" command.

Example:

(Ring the bell five times)

**GOSUB** RING5

(Ring it five more times)

**GOSUB** RING5

(What the hell, ring it five more times)

**GOSUB** RING5

(And end the procedure)

END

:RING5

BELL

BELL

BELL

BELL

BELL

(Enough ringy-dingy's, return back to the GOSUB that called "RING5")

RETURN

# RUN STR EXP

Description: This command is used to transfer from Red Ryder to a second Macintosh application, bypassing the Finder. STR\_EXP is the filename of the application to execute. If filename is wrong or doesn't exist, an error message is generated to the user and the Procedure File is terminated, but Red Ryder is not terminated. WARNING: The "RUN" command is not guaranteed to work properly in all Macintosh environments, such as those running under a file sharing or multitasking system.

Example:

(Here's a handy little Procedure I use to branch directly to another) (Macintosh application. It first restricts the GETFILE command to) (files of only type "APPL", which is the filetype of all Macintosh) (applications, has the user select the file, and the executes it.)

GETSELECT APPL

GETFILE A\$

(Make sure he didn't click the Cancel button)

IF NO END

IF YES RUN A\$

#### WAIT STR EXP

Description: The "WAIT" command is used to temporarily stop execution of a Procedure File until a certain time of day. STR\_EXP is the time of day to wait until in 24 hour format (10:00:00 PM would be given as 22:00:00). The time must be in the form:

HH:MM:SS

where:

HH = hours (00 to 24)

MM = minutes (00 to 59)

SS = seconds (00 to 59)

leading zeros are necessary for values less than 10. The string:

5:30:8

must be supplied with leading zeros, as in:

05:30:08

to function correctly. When the specified time is reached, execution resumes normally with the command following the "WAIT" command.

Example:

(Wait until 10:30 PM)

WAIT 22:30:00

(And ring a bell)

BELL

# User Defined Menus

Red Ryder gives you the opportunity to create your own customized pull down menu that can be used to either send a string of characters or execute a Procedure File. You have complete control of how your menu is named and it's contents, and your menu can contain up to 20 menu choices. Here is an example Procedure File that will create a menu called "Scott" which has five choices. The first four choices will just send a string telling us which menu choice sent it. The last menu choice will execute a Procedure File named "PROCFILE" which is on a disk named "HD20".

(Make sure there aren't any user defined menus left around by some) (other Procedure File)

#### MENUOFF

(Create a menu called "Scott" in Memory)

**DEFINE MENU** Scott

(Add the five menu choices I want in order)

ADD TO MENU This is menu choice #1

ADD TO MENU This is menu choice #2

ADD TO MENU This is menu choice #3

ADD TO MENU This is menu choice #4

ADD TO MENU This is menu choice #5

(Tell Red Ryder what each macro key is supposed to do)

MENUDOES 1, I just chose menu choice #1^M

MENUDOES 2, I just chose menu choice #2^M

MENUDOES 3, I just chose menu choice #3^M

MENUDOES 4,I just chose menu choice #4^M

MENUDOES 5, \HD20: PROCFILE

(Show the menu in the menu bar)

INSERT MENU

END

# **ADD TO MENU STR EXP**

Description: This adds a menu choice to the menu you've defined. The new choice is added to the end of any existing commands. There is no way to insert a command before others or delete individual commands, so choose the order of your menu choices with forethought. STR\_EXP contains the text that will appear in the menu choice with one notable exception. Certain special characters in STR\_EXP are used to affect the appearance of the menu choice, so be careful about using non-alphanumeric characters in STR EXP.

The following is provided for very experienced Macintosh users only. You will probably never use hardly any of the following information (except maybe the dividing line thingy), and some of it is pretty obtuse, but I'll document it here so you can play with it and show people all kinds of "betcha didn't know" magic later.

The three characters ^ ! and / should never be used in menu choice strings. Trust me on that. The three characters < ( and - do have special functions which are of some use. Here's what they do:

#### Character Function

<

The menu item has a special character style. The style to use is designated by the character following the ^ character and is one of the following:

<B - boldfaced

<I - italicized

<U - underlined

<O - outlined

<S - shadowed

I don't know that there's any reason more than one style couldn't be used, but let's keep this thing under control, eh?

Menu is filled with a line of hyphen characters. This should always be used with the (character so that the choice is not selectable by the user.

Choice is disabled. Many Mac menus have a dividing line of disabled hypen characters. To put one of those in your menu, you would use the special command:

ADD TO MENU (-

As another example, the command:

ADD TO MENU This is a choice<0<U

Would make the words "This is a choice" appear as both outlined and shadowed in the menu.

# DEFINE MENU STR EXP

Description: This gives your menu a name. I highly recommend you do a "MENUOFF" command before executing this command to be sure any previously defined menu is disposed of before the new one is defined. Note that this just creates the menu in memory — it does not display the menu in the menu bar. Once you've executed this command, you'll do some "ADD TO MENU" and "MENUDOES" commands to define the contents, appearance, and functions of the menu choices, and then execute an "ENABLE MENU" command to display the menu in the menubar and make it choosable by the user. STR\_EXP holds the menu's title that will appear in the menu bar when the menu is enabled, and it may not be more than 20 characters long.

# DISABLE MENU NUM EXP

\_\_\_\_\_

Description: This command can be used to disable (make gray and unselectable) any menu item (if NUM\_EXP is from 1 to 20) or the entire menu if NUM EXP is zero.

Example:

(Disable the third choice in my menu)

DISABLE MENU 3

#### ENABLE MENU NUM EXP

\_\_\_\_\_\_

\_\_\_\_\_\_

Description: After any DISABLE MENU command, this command can be used to enable any menu item (if NUM\_EXP is from 1 to 20) or the entire menu if NUM EXP is zero.

Example:

(Re-enable the third choice in my menu)

ENABLE MENU 3

## INSERT MENU

Description: This command is done after all of the menu items have been added to the menu, it simply draws the menu in the menu bar in an

# MENUDOES NUM EXP, STR EXP

\_\_\_\_\_\_

Description: This command defines exactly what your menu choices do when they are selected. NUM\_EXP is a number from 1 to 20 that corresponds to a menu choice (1 being the topmost menu choice and 20 being the bottommost menu choice). If the first character of STR\_EXP is a backslash (\), the rest of STR\_EXP is considered to be a filename of a Procedure File to execute immediately. If the first character is not a backslash, STR\_EXP it is considered to be a string of characters to send through the serial port immediately. In other words, STR\_EXP corresponds exactly to what you would put into a Macro Key string (including special and control characters). STR\_EXP may not exceed 40 characters.

# **MENUOFF**

Description: This disables the defined menu, removes it from the menu bar, and wipes it out of memory. If there is no defined menu, this command does nothing. User-defined menus are not removed or disabled when a Procedure terminates - they stay on the menubar until a "MENUOFF" command is executed. Therefore, it's a real good idea to put a "MENUOFF" command before any "DEFINE MENU" commands to wipe out any old user-defined menus left behind by other Procedure Files. Example:

(Get rid of my user defined menu) **MENUOFF** 

# User Defined Dialog Box Commands

Red Ryder's Procedure language allows you to use one of five preprogrammed dialog boxes for displaying information and collecting information from the user. Each of the five different dialog boxes has a different appearance, and is useful in a specific kind of user notification or input request.

#### QUERY1 STR VAR

Description: The dialog box brought up by this command contains three lines of prompting text, one editable text item, and an "OK" button. On entry, you should pass the three lines of text to prompt the user with in the string variables X\$, Y\$, and Z\$ (from top to bottom). On exit, STR\_VAR will contain the text the user typed into the editable text item.

Example:

(Put my prompting message in X\$, Y\$, and Z\$) COPYINTO X\$, Please type the name of your favorite music group, COPYINTO Y\$, "Hack And Slash" movie, or television evangelist. COPYINTO Z\$, If you have no favorite, just click the "OK" button. (Bring up the dialog box and get the user's input) QUERY1 A\$ (His response is now in A\$)

# **OUERY2**

Description: The dialog box brought up by this command contains three lines of prompting text and an "OK" button. On entry, you should pass the three lines of text to prompt the user with in the string variables X\$, Y\$, and Z\$ (from top to bottom).

Example:

(This just tells him what we're about to do, it doesn't get any info) (from the user)

COPYINTO X\$, This is going to take me a while.

COPYINTO Y\$, Go grab yourself some dinner, OK?

(We don't need three lines, so just erase Z\$ so it shows up blank)

ERASE Z\$

(And put up the dialog)

\_\_\_\_\_

QUERY2

# QUERY3

Description: The dialog box brought up by this command contains three lines of prompting text, a "Yes" button, and a "No" button. On entry, you should pass the three lines of text to prompt the user with in the string variables X\$, Y\$, and Z\$ (from top to bottom). On exit, the Yes/No flag will be set to "YES" if the user clicked on the "Yes" button, or "NO" if the user clicked on the "No" button. Example:

(Ask him if he really, really, etc. wants to do something) COPYINTO X\$, Do you really, really, really, COPYINTO Y\$, really, really, really, really,

COPYINTO Z\$, REALLY, REALLY want to hear the bell?

(Put up the dialog box)

\_\_\_\_\_\_

QUERY3

(If he clicked on the "No" button, quit the Procedure)

IF NO END

(Otherwise, we know he clicked on the "Yes" button) BELL

# QUERY4

Description: The dialog box brought up by this command contains one line of prompting text, a "Yes" button, and a "No" button. On entry, you should pass the line of text to prompt the user with in the string variable X\$. On exit, the Yes/No flag will be set to "YES" if the user clicked on the "Yes" button, or "NO" if the user clicked on the "No" button.

Example:

(Get a brief confirmation)

COPYINTO X\$, Wanna hear the bell?

(Put up the dialog box)

QUERY4

(And succumb to the user's command)

IF YES BELL

# QUERY5

Description: The dialog box brought up by this command contains one line of prompting text and an "OK" button. On entry, you should pass the line of text to prompt the user with in the string variable X\$. Example:

(Let the user know that everything is fine) COPYINTO X\$, The Procedure was successful! (Put up the dialog box) QUERY5

# Procedure File Examples

# **EXAMPLE #1:**

Description: Shows how the "@ ON", "@ OFF", and "SHOW@" commands are used together to display the Special Status Bar. Also used: PROMPT, ERASE, and COPYINTO.

(Load the three @ variables with our message to display)

COPYINTO @0, Waiting for the string "CONNECT" to arrive.

COPYINTO @1, Please be patient!

(We only need two lines, so we'll ERASE the third)

ERASE @2,

(Turn on the Special Status Bar)

@ ON

(And show the contents of the three @ variables)

SHOW@

(Now wait for the "CONNECT" string to arrive)

PROMPT CONNECT

(And turn off the Special Status Bar)

@ OFF

#### **EXAMPLE #2:**

Description: Shows how the "ALERT" and "PROMPT" commands are used together to look for two different possible strings to be received, and react by branching to the correct routine to handle the one that is received. Also used: **BELL, END,** and **JUMPTO**.

(Wait for either "you have mail" or the string "Command?" and branch to) (appropriate routine. This allows me to be looking for two things at) (once)

ALERT1 you have mail/JUMPTO HAVEMAIL

PROMPT Command?

(I got the "Command?" string so I JUMPTO somewhere else)

JUMPTO DOCOMMAND

:HAVEMAIL

(If I get here, I got the "you have mail" string and need to)

(insert the commands here to handle that)

(But for this example, we'll just string 'em both together)

: DOCOMMAND

BELL

END

# **EXAMPLE #3:**

Description: Shows how the "USEROPENO", "USEROPENA", "USEROPENI", "USERREAD", "USERWRITE", "USERWRCR", and "USERCLOSE" commands are used to create, read, and write data with disk files. The Procedure creates a file, writes two lines of text, closes the file,

```
reopens the file, writes one more line of text, closes the file, reopens the file for
input, reads the lines of text back in, and finally closes the file again.
(Create the file for output using path number 1 on a disk)
(named "MYDISK")
USEROPENO 1, MYDISK: DATAFILE
(Write a line of text to the file)
USERWRITE 1, This is line #1
(Terminate it with a carriage return)
(And do the same thing again for the second line)
USERWRITE 1, This is line #2
USERWRCR 1
(Now close the file)
USERCLOSE 1
(Let's reopen it now for append)
USEROPENA 1, MYDISK: DATAFILE
(And write one more line of text to it)
USERWRITE 1, This is line #3
USERWRCR 1
(And close it up again)
USERCLOSE 1
(OK, Let's open it up for input this time)
USEROPENI 1, MYDISK: DATAFILE
(And read the three lines of text into the string)
(variables A$, B$ and C$)
USERREAD 1,A$
USERREAD 1,B$
USERREAD 1,C$
(A$ now contains "This is line #1")
(B$ now contains "This is line #2")
(C$ now contains "This is line #3")
(Close up the file to finish things off)
USERCLOSE 1
```

## Example #4:

I want to call Big Bob's BBS using 1200 Baud, NO Parity, 8 Data Bits, 1 Stop Bit, and FULL Duplex. The number is (000)555-1212. I'm using a Hayes compatible modem. Please note that this is a bogus phone number.

```
(Set the serial port settings)

COMM 1200-N-8-1-FULL

(Tell the modem to dial the number)

DIAL ATDT 1 000 555 1212
```

# Example #5:

Here's how I log onto CompuServe (using a CIS node - not Telenet!). I have a Macro Keys file named CIS MACROS on the disk named MYDISK that I want to load in, also.

```
(Set the communications parameters)
COMM 1200-N-8-1-FULL
(Load in my Macro Keys file - you may or may not want to do this)
MACRO MYDISK: CIS MACROS
(Now dial my local access number)
DIAL ATDT 555-1212
(Wait for the last part of the message CONNECT from the modem)
PROMPT NECT
(reset my elapsed time and billing clocks so I know how long)
(I've been on and how much I've spent)
(hang loose a second for the connection to get established solidly)
PAUSE
(need to send them a CONTROL-C to get started)
TYPE ^C
(CompuServe asks me for "User ID:")
PROMPT ID
(Always use a PAUSE after a PROMPT!)
PAUSE
(Type in my user ID with a carriage return at the end)
TYPE 73176,61^M
(Now CIS asks me "Password:")
PROMPT word
(Give the line a chance to settle down)
PAUSE
(I type in my password now, followed by a carriage return)
TYPE MY/SECRET^M
(The ! character is what CIS uses for a command prompt)
PROMPT !
(Getting the hang of this PROMPT/PAUSE routine, now?)
PAUSE
(Onward to MAUG's MAC USER forum!)
TYPE GO MACUS^M
(Hey human, ol' Red did all the work! Wake up! Wake up!)
BELL
BELL
BELL
BELL
```

# Example #6:

I've just downloaded a neat public domain program called Space Potatoes, put it on a disk called GAMES and three of my buddies want a copy. I just instuct my modem to answer the phone (using the "ATS0=1" modem command), tell them to give me a call, type in the secret password GOOBER, and my Mac will send the file XMODEM. I'm going to put this Procedure on that same disk and call it BUDDIES PROCEDURE. Of course, while one guy's online, the other two have their Red Ryder's set up for redialing.... Remember that in order for this to work, all of your buddies have to call in using 1200-N-8-1-FULL

# parameters.

```
(Set up the serial port settings)

COMM 1200-N-8-1-FULL

(I can ignore everything else until my secret password comes in!)

PROMPT GOOBER

(Tell 'em what we're about to do - notice the leading carriage)

(return to get past the password they typed and make things look clean)

TYPE ^MStarting XMODEM transfer^M

(Send that file, Red!)

SENDX GAMES:SPACE POTATOES

(Start this procedure over again for whoever's left.)

DO GAMES:BUDDIES PROCEDURE
```

# Example #7:

Here's my super-simple "Hang Up The Phone" procedure that I like to link through a Macro Key. Works only with Hayes and compatible modems.

```
(Type the modem escape sequence - NO CARRIAGE RETURN HERE!)

TYPE +++

(Modem will go into command state and return "OK" message)

PROMPT OK

(So I now pause a moment and type the modem hang up command with a)

(carriage return)

PAUSE

TYPE ATH^M

(Modem responds again with "OK" and disconnects phone)
```

# Example #8:

I like to selectively turn on and off the capturing of certain information to my disk for later archiving and perusal. By taking advantage of the fact that the "RECA" Procedure command does not destroy a file if it exists (like the Capture Incoming Data To TEXT File command under the File menu) but appends the new data to the end of the existing file, I can easily come up with a "Capture On/Capture Off" pair of Procedure Files that can be linked into Macro Keys for one-click operation. I want the capture file to be named "Stored Data" on the disk named "Data Disk" and the following Procedures will be on the disk named "Red Ryder"

1) Create and compile the following procedure and save it on the disk "Red Ryder" with the name "Capture On".

#### QUIET

RECA Data Disk:Stored Data

2) Create and compile the following procedure and save it on the disk

"Red Ryder" with the name "Capture Off".

QUIET CLOSE

3) Put the following string into Set #1, Macro Key #8

\Red Ryder:Capture On

4) Put the following string into Set #1, Macro Key #9

\Red Ryder:Capture Off

Give those two Macro Keys appropriate labels, while you're at it, ("Capture On" and "Capture Off: would be just fine).

Now, whenever you select Macro Key #8 from Set #1, incoming data will be directed to the file "Stored Data" on the disk "Data Disk". Select Macro Key #9 from Set #1 to turn off this saving of incoming data. When you have the Macros Status Bar on the screen, you can turn the capturing on and off with a click of your mouse!

Example #9:

I want to use the Host Mode on my office computer when I get home from work, but I don't want anyone else being able to get in except me. The following Procedure waits for the secret password "GAZORK" before letting me into the Host Mode. I'm going to name this Procedure "Password" on the disk "MYDISK". Note: remember to set your modem to auto-answer the phone by using the modem command "ATSO=1".

(The following message won't be seen when you start up this Procedure) (the first time - it's just there to let you know that the Procedure) (is running again when you later exit the Host Mode)

TYPE Password protect installed. Hang up now...^M

(The modem should already be in auto-answer mode, so we'll just wait) (for our magic password to come in and ignore everything else)

PROMPT GAZORK

(Caller is now let in. Tell them what we're about to do and what to do) (before disconnecting)

TYPE Now going to Host Mode. Type in the command^M

TYPE "DO MYDISK: Password" when you're done^M

(And now quit the Procedure and enter the Host Mode)  ${f HOST}$ 

Example #10:
Redialing one service is fine, but let's get real clever and redial 4 services (or more if you wanted) in a daisy-chain fashion. When you get connected, Red

rings like crazy till you choose Cancel Procedure from under the Service menu. This procedure assumes that the communications parameters are the

same for all four services, and have been set before this Procedure is executed. The two ALERT commands are used to trap the modem's "NO CARRIER" and "BUSY" messages, and the PROMPT is used to trap the "CONNECT" message. Suggestion: you could dress this up by inserting the name of the service you are currently dialing into a string variable, and then use the CONCAT command to attach the name of the connected service to the displayed message.

```
:FIRST
DIAL ATDT 555-1111
ALERT1 CARR/JUMPTO SECOND
ALERT2 BUSY/JUMPTO SECOND
PROMPT NECT
JUMPTO ONLINE!
: SECOND
PAUSE
DIAL ATDT 555-2222
ALERT1 CARR/JUMPTO THIRD
ALERT2 BUSY/JUMPTO THIRD
PROMPT NECT
JUMPTO ONLINE!
: THIRD
PAUSE
DIAL ATDT 555-3333
ALERT1 CARR/JUMPTO FOURTH
ALERT2 BUSY/JUMPTO FOURTH
PROMPT NECT
JUMPTO ONLINE!
PAUSE
: FOURTH
PAUSE
DIAL ATDT 555-4444
ALERT1 CARR/JUMPTO FIRST
ALERT2 BUSY/JUMPTO FIRST
PROMPT NECT
:ONLINE!
COPYINTO @0, Connected to remote service!
COPYINTO @1, Please select "Cancel Procedure" from under the Service
COPYINTO @2, menu to go online.
@ ON
SHOW@
:LOOP
BELL
PAUSE
JUMPTO LOOP
```

# Example #11:

When you change your communications parameters, your Hayes compatible

modem may or may not be ready to immediately respond to further commands. What you need to do first is get the modem to recognize your new parameters by typing the "attention" modem command "AT" a few times until your modem responds with the message "OK". The following routine should be used after any COMM command.

(Set new serial port settings)

COMM 1200-N-8-1-FULL

(Send "AT" command and carriage return to get modem's attention)

TYPE AT^M

PAUSE

(Send it again to make sure it's all set)

TYPE AT^M

PAUSE

(Yes Virginia, we are going on blind faith that the "OK" really came)

(back from the modem, but I've never had this fail me)

(If you're REALLY paranoid, you could use ONPANIC and PANICAFTER)

(commands with a PROMPT OK to be sure that the "OK" actually came back)

DIAL 555-1212

(Rest of Procedure goes here)

# Example #12:

The following code fragment can be added to the beginning of a Procedure File to make sure that only those who know the password (in this case, the word "SECRET") can execute it. If the Procedure is compiled with encryption, a routine like this can make the file impervious to compromise.

```
(Present a dialog box asking for a password)
COPYINTO X$, This procedure is limited to authorized users
ERASE Y$
COPYINTO Z$, Please type in your password
QUERY1 A$
(See if the user actually typed something)
EMPTY A$
IF YES JUMPTO ABORT
(Convert what he typed to all uppercase)
CONVUP A$
(See if it contains the password "SECRET")
CONTAINS SECRET
IF NO JUMPTO ABORT
JUMPTO CONTINUE
(This is what happens if he fails to type the correct password)
END
:CONTINUE
(Rest of Procedure begins here)
```

# Letting Red Ryder Write A Procedure File For You

The Write A Procedure For Me menu command under the Service menu can be used to write Auto-logon Procedures, Procedures to be used while inside a service, or even full session unattended Procedures. It basically monitors incoming and outgoing data and generates the proper Procedure commands to navigate in a mirror image fashion of your actions. Of course, this presumes that the host will react in a consistent manner from session to session. If it doesn't, a small amount of fine tuning may be necessary.

After selecting the Write A Procedure For Me command, you will be prompted to enter the resulting filename for the Procedure. It'll be saved as a TEXT file of Procedure command source code that you'll later need to compile. After specifying the name, you'll be asked whether or not you wish the Procedure to begin by intializing the system to all current terminal settings. If you answer affirmatively, all of the Procedure commands (such as COMM, SLOW ON, CONTROL1, etc) applying to settings parameters will be generated. It will also include a short sequence for initializing a Hayes compatible modem:

TYPE AT^M PAUSE TYPE AT^M PAUSE TYPE AT^M PAUSE

This matches baud rates and prepares the modem for a dialing command (you can edit out this sequence or modify it appropriately if you don't have a Hayes compatible modem). You should answer Yes to this prompt only if your Procedure will begin by dialing a service. For procedures that will be executed while already inside of a service, you would not want all of the initialization stuff to occur.

For this reason, you must have <u>all</u> settings the way you want them before you choose the **Write A Procedure For Me** command. Once the Procedure generation has begun, clicking on the baud rate (for instance) parameter will **not** be recorded in the Procedure file. Some menu command selections are supported during Procedure generation, they're listed below.

A good number of people (like me) have gotten into the shortcut habit of just typing the modem dialing command, rather than by using the **Dial Or Redial**A Number menu command. This habit must be broken for use with the Write

A Procedure For Me command. It is absolutely essential that you use the

Dial Or Redial A Number menu command for dialing a service when

generating a Procedure, or the resulting Procedure will most likely not work properly.

The Write A Procedure For Me command saves uninterrupted typing sequences, and then generates TYPE Procedure commands for the sequence when it's interrupted. A typing sequence is interrupted by a carriage return, either incoming from the host, or outgoing from the keyboard. If it's outgoing from the keyboard the letters 'M are appended to the end of the TYPE command to show that you typed a RETURN. When it's interrupted, Red Ryder looks at the line you were typing on to determine whether you were typing in response to a prompt from the host. If your typing began on the first column of the line, no PROMPT statement is generated. Otherwise, up to 10 characters preceding the first character you typed are used to generate a PROMPT command (which is followed by a PAUSE command).

The following menu command actions will be saved in the Procedure file:

- Load Macro Keys From Disk
- File receives
- Manual closing of an ASCII file receive.
- · File sends
- Print TEXT File
- Delete A File
- Echo Incoming Data To Printer
- Load Settings From Disk
- Dial Or Redial A Number

When you choose one of the above menu command actions, Red Ryder will again look at the most recent line received, if the line is empty, the appropriate Procedure command for the Menu choice will be placed immediately after the last command in the Procedure file. If the line isn't empty, Red Ryder will want to know if you did that menu action in response to a prompt from the host.

Let's look at an example of both methods. Before you used a **Dial Or Redial**A Number menu command, you might want to load in a set of Macro Keys from a file. Since you didn't do that menu command in response to any prompt from a host (you aren't even connected to a host yet!), you wouldn't want a **PROMPT** command generated before the menu action command.

However, opening a receive file to capture all new messages probably would be in response to a prompt from the host, and you would want such a **PROMPT** generated in the Procedure file.

# **Hints And Tips**

1) The Write A Procedure For Me menu choice turns into Stop Creating The Procedure while Procedure generation is active. Choose it to finish

generating the Procedure file.

- 2) While generating a Procedure file, type slowly and deliberately. Typing errors, backspacing, etc. will be repeated verbatim for the next billion years (or however long you continue to use that Procedure).
- 3) When a menu command and typing action are to take place after the same prompt from the host (like opening a receive file and then typing the command to list all new messages), choose the menu action first if possible (so that the **PROMPT** command will be placed before the menu action command, followed immediately by the typing sequence). A typing sequence must never be interrupted by a menu command action do one before or after the other only. However, if you're going to be sending or receiving a file using Kermit or XMODEM, you'll find that the keyboard is locked out after the menu command is executed, so in this case the typing sequence would come first, followed by the menu action.
- 4) The logic for this command was derived by observation, and there's always room for improvement and added utility. Please pass along any comments or suggestions to me so I can incorporate them in future versions.
- 5) Break signals, Macro Keys, VT100 status bar and numeric keypad sequences, and automatic VT100 cursor key positioning are not supported in Procedure generation). Typing sequences are limited to normal ASCII alphanumeric and control keys.

# Why Would I Want To Edit A Generated Procedure?

CompuServe is a good example of a inconsistent host. The first menu you may see after logging on is unpredictable - it may be either the top level command menu or the electronic mail menu if you have messages waiting. On the dark side, the prompts for these two menus are quite different, so a generated Procedure may get "hung up" because it expects the prompt for a menu that never arrives. On the bright side, since both menus use an exclamation point ("!") as the last character in the prompt, the generated Procedure can be easily edited to look for only that character in the first menu **PROMPT** command if all you wish to do is branch directly to another part of CompuServe,

# The GETPARAM And PUTPARAM Procedure Commands

RED FLAG NOTE: The information in this chapter is meant only for very experienced Red Ryder users!

The **GETPARAM** and **PUTPARAM** Procedure commands allow you to get at the "guts" of Red Ryder to perform operations normally not possible through other Procedure commands. The **GETPARAM** and **PUTPARAM** Procedure commands use the following syntax:

CETPARAM NUM\_EXP, NUM\_VAR PUTPARAM NUM\_EXP1, NUM\_EXP2

These commands are used to examine or modify one of many internal memory parameters used by Red Ryder. These internal parameters are referenced by a number which corresponds to their relative offset from the beginning of the default Settings File. The **GETPARAM** command is supplied with the internal parameter number in NUM\_EXP, and the contents of that internal parameter (which is always numeric) is placed in the numeric variable designated by NUM\_VAR. In other words, to get the current value in the internal parameter #100 and put it into the numeric variable A%, you would use the Procedure command:

GETPARAM 100, A%

The **PUTPARAM** command is used to modify the contents of an internal parameter (designated by NUM\_EXP1) with the contents of NUM\_EXP2. To change the contents of internal parameter #100 to a value of 15, you would use the Procedure command:

**PUTPARAM** 100,15

Because the **PUTPARAM** command modifies things that were never originally meant to be diddled by the user (by I have yielded for the benefit of those who understand how to handle digital dynamite), it is inherently dangerous. Never change an internal parameter not listed below, and modify those only with the values designated in the list. Perhaps this warning is making a mountain out of a molehill, but the problem is that these internal parameters are saved in Settings Files, and a wrong value can therefore propagate to other Settings Files without warning. If you plan to distribute Procedures that use the **PUTPARAM** instruction, please give thought to a beginner and display a proper warning message to the user before modifications are done (and please give him a chance to bail out if he gets scared!).

Think of it this way. Using the PUTPARAM command is like using ResEdit on a System file. If you understand that and the implications of making a mistake, don't be afraid to plunge ahead.

When there is an equivalent Procedure command, use that in favor of a PUTPARAM command, as the equivalent is guaranteed to be safer. For some things, like finding out what baud rate or duplex is being used at a given moment, there are no equivalent Procedure commands and a GETPARAM command must be used. Some operations (like defining the "Enter" key on an original Mac 512K keyboard as an Escape key) can only be done with the PUTPARAM command. You can use the GETPARAM command on any of the internal parameters listed below, but unless a listing below specifically states "Use PUTPARAM to modify", don't use the PUTPARAM command with it. By the way, you will find "holes" in the list below. Some internal parameters are of absolutely no interest to you, or are reserved for future expansion.

You'll probably never need any of this stuff. However, I have found a few things very useful, and if I've learned anything in the last 9 versions of Red Ryder, it's not to second guess your hunger for information and clever ways to use it. The following is presented with the forewarning that you need to know Red Ryder inside and out to make sense of it, and programming experience (when high and low bytes are discussed) may be necessary to use a few of these.

```
Parameter #0: Current baud rate. Use COMM command to modify.
                    0 = 300, 1 = 450, 2 = 1200, 3 = 2400, 4 = 4800,
Meaning:
                    5 = 9600, 6 = 19200, 7 = 57600
Parameter #1: Current parity. Use COMM command to modify.
Meaning:
                    0 = \text{None}, 1 = \text{Odd}, 2 = \text{Even}, 3 = \text{Mark}, 4 = \text{Space}
Parameter #2: Current Databits. Use COMM command to modify.
Meaning:
                    0 = 7 bits, 1 = 8 bits, 2 = 5 bits, 3 = 6 bits
Parameter #3: Current stopbits. Use COMM command to modify.
                    0 = 1 bit, 1 = 2 bits, 2 = 1.5 bits
Meaning:
Parameter #4: Current duplex. Use COMM command to modify.
                    0 = Full, 1 = Half, 2 = Echo, 3 = Null
Meaning:
Parameter #6: XMODEM and initial Kermit timeout in seconds. Use
   PUTPARAM to modify.
                    1 - 255 (seconds)
Meaning:
Parameter #7: ASCII char for control button #1 on General Status Bar.
Parameter #8: ASCII char for control button #2 on general status bar
Parameter #9: ASCII char for control button #3 on general status bar
   Use CONTROL1, CONTROL2, or CONTROL3 commands to modify.
```

Meaning: (0-255) ASCII value of character button sends.

Parameter #10: Current number of screen columns. Use **DISPLAY** command to modify.

Meaning: (20-132) Number of columns

Parameter #11: Current terminal emulation type. Use TTY, VT52, or VT100 commands to modify.

Meaning: 0 = TTY, 1 = VT100, 2 = VT52

Parameter #13: Current Status Bar. Use GBAR, BBAR, MBAR, VBAR, LOUD, or QUIET commands to modify.

Meaning: 1 = General, 2 = Buffered Keyboard, 3 = Macros,

4 = VT100, 5 = Monitoring procedure

Parameter #14: Should cursor flash? Use PUTPARAM to modify.

Meaning: 0 = no flash, 1 = flashing

Parameter #15: Cursor shape. Use PUTPARAM to modify.

Meaning: 0 = underline, 1 = block

Parameter #16: Amount of delay between each character (in 60ths of a second) sent with a **SENDA** Procedure command or **Send TEXT File** menu command if Parameter #27 is nonzero and Parameter #32 = 1. Use **PUTPARAM** to modify.

Meaning: (0 - 255) number of 60th's of a second to delay.

Parameter #17: Function of Enter key on original Macintosh 512K (non-enhanced and without a built-in numeric keypad) keyboard. Use **PUTPARAM** to modify only if you know that such a keyboard is attached.

Meaning: 0 = Same as Return key, 1 = ESCAPE key (Parameter #98 overrides this)

Parameter #18: Backspace key mapping. Use **DELKEY** command to modify. Meaning: 0 = backspace (ASCII 8), 1 = DELete (ASCII 127)

Parameter #19: Active set when macros status bar is showing, or when Macro Key is executed from keyboard (using COMMAND-number). Use MBAR command to modify.

Meaning: 1, 2, or 3 (active set number)

Parameters #20 through 23: Four ASCII characters to use as a file type (default MACA) for TEXT file or non-MacBinary receives. Use **PUTPARAM** to modify.

Meaning: Each parameter contains ASCII value of character.

Parameter #24: Use auto-receive file naming convention? Use **PUTPARAM** to modify.

Meaning: 0 = no, get name first, 1 = receive file first, use MacBinary name if possible or ask for name then.

Parameter #25: Send text files as MacBinary? Use **PUTPARAM** to modify. Meaning: 0 = no, 1 = yes

Parameter #26: Before sending, instruct remote to auto-receive? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #27: Delay after each character or line (depending on Parameter #32) sent with a **SENDA** Procedure command or **Send TEXT File** menu command. Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #28: Should file sent with **SENDA** Procedure command or **Send TEXT File** menu command be forcefully wrapped at a certain column?
Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #29: Column number to wrap line if Parameter #28 is nonzero. Use **PUTPARAM** to modify.

Meaning: (20-132) column number

Parameter #32: Kind of delay for **SENDA** procedure command or **Send TEXT File** menu command if Parameter #27 is nonzero. Use **PUTPARAM** to modify.

Meaning: 0 = delay between lines sent, 1 = delay between characters sent

Parameter #33: Enable CompuServe 'B' protocol recognition? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #34: Supercharged XMODEM? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #35: Use 1K blocks for XMODEM? Use PUTPARAM to modify.

Meaning: 0 = no, 1 = yes

Parameter #36: Enable RLE graphics recognition? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #37: Scroll lines into saved screens buffer before full screen clear? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Persenter #38: Hermit was special ^Q line turnaround handshake? Use SUFFERENT to modify.

Meaning:

0 = no, 1 = yes

Parameter #39: Should VIMouse delay? Use PUTPARAM to modify.

Meaning:

0 = no, 1 = yes

Parameter #40: Should 'key send the 'character or the ASCII code in Parameter #145? Use ALT command to modify.

Meaning:

0 = send ` character, 1 = send ASCII code in

Parameter #145.

Parameter #41: Should Procedures be monitored in status bar? Use LOUD and QUIET commands to modify.

Meaning:

0 = no, 1 = yes

Parameter #42: Should date/time heading be printed on the printer? Use **PUTPARAM** to modify.

Meaning:

0 = no, 1 = yes

Parameter #43: If Parameter #42 is nonzero, should date/time be printed on every page or first page only. Use **PUTPARAM** to modify.

Meaning:

0 = every page, 1 = first page only

Parameter #44: Should printer form feed when the number of lines in Parameter #45 has been printed? Use **PUTPARAM** to modify.

Meaning:

0 = no, 1 = yes

Parameter #45: If Parameter #44 is nonzero, do form feed after this number of lines. Use **PUTPARAM** to modify.

Meaning:

(1-255) Number of lines before form feed

Parameter #46: Wait for character in Parameter #488 after each line sent by SENDA procedure command or Send TEXT File menu command? Use PUTPARAM to modify.

Meaning:

0 = no, 1 = yes

Parameter #48: Is data currently being echoed to the printer? Use ECHO command to modify.

Meaning:

0 = no, 1 = yes

Parameter #49

Parameter #50: These two bytes contain the cost(in 100ths of a cent) for the billing clock to increment after each elapsed minute. It is a two byte integer value, so Parameter #49 holds the high byte value, and Parameter #50 holds the low byte value. Use SETCOST to modify.

Parameter #60 = Should the following functions include blank lines in their output? Use **PUTPARAM** to modify.

- 1) "Archive Display Screen" menu command
- 2) "Archive All Screens" menu command
- 3) Screen To Disk Button
- 4) Screen To Printer Button
- 5) SCREENDISK Procedure command.
- 6) SCREENPRINT Procedure command.

Meaning:

0 = yes, 1 = no

Parameter #61: Should the **Delete A File** menu choice skip the confirming dialog after a file is chosen? Use **PUTPARAM** to modify. Meaning: 0 = no, 1 = yes

Parameter #62: Should the Delete A File menu choice continually cycle through the select and delete process until the user clicks on the file selection dialog's "Cancel" button? Use **PUTPARAM** to modify. Meaning: 0 = no, 1 = yes

Parameter #63: Should the **Find Text** menu choice use a zooming rectangle when the window scrolls? Use **PUTPARAM** to modify. Meaning: 0 = no, 1 = yes

Parameter #64: Should Macros Status Bar scroll into next (or previous set) or just immediately display it when selected by mouse in the scroll bar? Use **PUTPARAM** to modify.

Meaning: 0 = scroll, 1 = jump

Parameter #65: Display title screen on program startup? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #66: Should "Send screen to printer" button on General Status Bar confirm with a dialog box that the user actually wants to do that? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #67: Should "Archive display screen" button on General Status Bar confirm with a dialog box that the user actually wants to do that? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #68: Should close box in Terminal Display Window confirm with a dialog box that the user actually wants quit? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #69: Should "Reset time and billing clocks" button on General Status Bar confirm with a dialog box that the user actually wants

to do that? Use PUTPARAM to modify.

Meaning:

0 = yes, 1 = no

Parameter #70: If a Procedure File is executed by a macro key while another Procedure File is executing, should we put up a dialog telling the user to cancel the executing Procedure, or should we cancel it automatically and execute the new Procedure? Use PUTPARAM to modify.

Meaning:

0 = dialog, 1 = auto-execute

Parameter #73: Maximum packet size Red Ryder's Kermit wants (default is 94). Use **PUTPARAM** to modify.

Meaning:

(1 - 94) maximum packet size in bytes.

Parameter #74: Kermit's start of header character (default is 1). Use PUTPARAM to modify.

Meaning:

(0-255) ASCII code for start of header character.

Parameter #75: Kermit's definition of SPACE character (default is 32).
Use **PUTPARAM** to modify.

Meaning:

(0-255) ASCII code for SPACE character.

Parameter #76: Kermit's definition of CR character (default is 13).
Use **PUTPARAM** to modify.

Meaning:

(0-255) ASCII code for CR character.

Parameter #77: Kermit's definition of DELETE character (default is 127).
Use **PUTPARAM** to modify.

Meaning:

(0-255) ASCII code for DELETE character.

Parameter #78: Kermit's definition of control character QUOTE character (default is 35, a # character). Use **PUTPARAM** to modify. NOTE: Red Ryder will always use the '&' character for 8th bit quoting when anything but NO parity is used.

Meaning:

(0-255) ASCII code for QUOTE character.

Parameter #79: Maximum number of retries Kermit will do before aborting a file transfer (default is 5). Use **PUTPARAM** to modify.

Meaning:

(1 - 255) maximum number of retries

Parameter #80: Number of padding characters Red Ryder's Kermit wants (default is 0). Use **PUTPARAM** to modify.

Meaning:

(0 - 255) number of padding characters

Parameter #81: ASCII code of padding character Red Ryder's Kermit wants (default is 0). Use **PUTPARAM** to modify.

Meaning:

(0 - 255) ASCII code of padding character

Parameter #82: EOL character Red Ryder's Kermit wants (default is 13), Use **PUTPARAM** to modify.

Meaning: (0 - 255) ASCII code of EOL character.

Parameter #83: Number of seconds Red Ryder's Kermit wants remote Kermit to wait before timing out (default is 3). Use **PUTPARAM** to modify. Meaning: (1 - 255) seconds to wait

Parameter #84: Should **TEXT File Transfer Preferences** menu choice's "Strip Control Characters" option allow tabs to pass through to the file in addition to carriage returns? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #85: Do form feed after printing finishes? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #86: Ignore TTY full-screen clears (ASCII 12)? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #87: Buffered keyboard right margin column. Use **PUTPARAM** to modify.

Meaning: (1 - 255) right margin column number.

Parameter #88: Current font number. Use FONT command to modify.

Meaning: (0 to 255) font number

Parameter #89: Current font size. Use FONT command to modify.

Meaning: (1 to 255) font point size

Parameter #90: High byte of maximum symbol definitions.

Parameter #91: Low byte of maximum symbol definitions. Use **PUTPARAM** to modify either of these.

Parameter #92: High byte of maximum symbol references.

Parameter #93: Low byte of maximum symbol references. Use **PUTPARAM** to modify either of these.

Parameter #94: Does the compiler encrypt Procedure Files? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #95: Should the high bit (parity bit) be zapped out from incoming data? Use NOZAP command to modify.

Meaning: 0 = zap, 1 = don't zap

Parameter #96: Show compiler window during compilation? Overridden by

**COMPILE LOUD** and **COMPILE QUIET** commands. Use **PUTPARAM** to modify.

Meaning:

0 = no, 1 = yes

Parameter #97: Should the DTR line be dropped when exiting Red Ryder?
Use PUTPARAM to modify.

Meaning:

0 = drop DTR, 1 = don't drop DTR

Parameter #98: Should the Enter key on the original Macintosh 512K (not enhanced with no built-in numeric keypad) keyboard send a linefeed?

Use PUTPARAM to modify.

Meaning:

0 = no, 1 = yes

Parameter #142: Do we have an additional VT100 font? Use **PUTPARAM** to modify.

Meaning:

0 = no, 1 = yes

Parameter #143: If Parameter #142 is nonzero, additional VT100 font high byte. Use **PUTPARAM** to modify.

Parameter #144: If Parameter #142 is nonzero, additional VT100 font low byte. Use **PUTPARAM** to modify.

Parameter #145: If Parameter #40 is nonzero, this contains the ASCII code that that the 'key will send. Use **PUTPARAM** to modify.

Meaning: (0 - 255) ASCII code of character to send

Parameter #146: Override default VT100 special keys mapping and use settings installed by "Install Keypad" utility? Use "Install Keypad" utility to modify.

Meaning:

0 = use default, 1 = use installed

Parameter #488: ASCII character to wait for after each line sent with a SENDA Procedure command or Send TEXT File menu command if Parameter #46 is nonzero. Use PUTPARAM to modify.

Meaning:

(0 - 255) character to wait for

Parameter #489: Amount of delay after each line (in seconds) sent with a SENDA Procedure command or Send TEXT File menu command if Parameter #27 is nonzero and Parameter #32 is zero. Use PUTPARAM to modify.

Meaning:

(1 - 255) delay time in seconds

Parameter #492: Current serial port in use. Use MODEM and PRINTER commands to modify.

Meaning:

0 = printer, 1 = modem

Parameter #493: Should the Return key send a carriage return only or a carriage return followed with a linefeed? Use LF command to modify.

Meaning: 0 = carriage return only, 1 = CR/LF

Parameter #494: Strip control characters from XMODEM non-Macbinary

receives? Use XKSTRIP command to modify.

Meaning:

0 = no, 1 = yes

Parameter #495: Recognize and use MacBinary for non-TEXT files? Use

**PUTPARAM** to modify.

Meaning:

0 = no, 1 = yes

Parameter #503: Use and recognize CRC error checking? Use CRC command

to modify.

Meaning:

0 = yes, 1 = no

Parameter #504: Is a Macintosh Plus keyboard (with built-in numeric

keypad) currently in use? DO NOT MODIFY!

Meaning:

0 = no, 1 = yes

Parameter #505: Redial attempt limit. Use REDIAL LIMIT command to

modify.

Meaning:

(0 - 255) attempts

Parameter #506: Strip control characters from RECA Procedure commands or Capture Incoming Data To TEXT File menu commands? Use

STRIP command to modify.

Meaning:

0 = no, 1 = yes

Parameter #507: VT100 relative origin mode. Use PUTPARAM to modify.

Parameter #508: VT100 smooth scroll mode. Use PUTPARAM to modify.

Meaning: 0 = reset, 1 = set

Parameter #509: VTMouse waits for host? Use PUTPARAM to modify.

Meaning: 0 = no, 1 = yes

Parameter #585: VT100 Wraparound mode. Use PUTPARAM to modify.

Parameter #586: VT100 Autorepeat mode. Use PUTPARAM to modify.

Parameter #587: VT100 Newline mode. Use PUTPARAM to modify.

Parameter #588: VT100 Cursor key mode. Use PUTPARAM to modify.

Parameter #589: VT100 Keypad mode. Use PUTPARAM to modify.

Meaning: 0 = reset, 1 = set

# **Index Of Menu Commands**

# File Menu

Capture Incoming Data To TEXT File: 32, 39, 51, 52, 63, 64, 96, 127

Receive File - XMODEM: 54, 57, 58, 98

Receive File - Kermit: 58, 97

Receive Batch - YMODEM: 57, 58, 98 Send TEXT File: 9, 46, 47, 49, 59, 67, 98

Send File - XMODEM: 49, 54, 98

Send File - Kermit: 54, 98 Send Batch - YMODEM: 57, 98 Create YMODEM Batch: 57, 98

Delete A File: 62, 63, 132 Set Auto-Receive Volume: 57

Set Archive Volume: 9

Quit: 6, 116

# Edit Menu

Undo: 10 Cut: 22 Copy: 8

Paste: 9, 22, 99 Clear: 10, 22 Copy Table: 8, 9

Print Selected Text: 8, 9, 38 Append To Archive File: 8

Find Text: 8, 63

Find Same Text Again: 8
Edit Control Buttons: 17
Edit Macro Keys: 25

## Service Menu

Initiate Procedure: 26, 73

Write A Procedure For Me: 131, 132 Compile TEXT File to Procedure: 74

Check Procedure Syntax: 74

Dial Or Redial A Number: 40, 44, 45, 84, 108, 131, 132

Monitor Procedure: 73, 81, 106, 108, 111

Set Billing Cost: 16, 109 New Phonebook: 42 Open Phonebook: 42

#### Local Menu

General Status Bar: 16, 24, 38, 64, 66, 101, 103, 104, 106

Buffered Keyboard Status Bar: 22, 86

Macros Status Bar: 24, 25, 63, 64, 73, 76, 91, 107, 128

VT100 Status Bar: 22, 28, 35, 37, 110, 133

Clear Display Screen: 8 Clear All Screens: 7, 8

Archive Display Screen: 9, 16 Archive All Screens: 9, 63

Print TEXT File: 9, 38, 39, 103, 132

Echo Incoming Data To Printer: 38, 39, 63, 103, 132

Tab Stops: 10, 11, 22

Host Mode: 12, 13, 66, 67, 68, 69, 73, 112, 128

Bring Other Open Windows To Top: 6

Hide Terminal Window: 6, 108

Change Serial Port Settings: 15, 17, 68

Set Font: 10, 71, 105

# Customize Menu

XMODEM And Kermit Preferences: 57, 58, 61, 104, 109, 110 TEXT File Transfer Preferences: 47, 48, 49, 52, 60, 64, 109

Terminal Emulation Preferences: 32, 34, 35, 105

Keyboard Mapping Preferences: 18, 19, 47, 66, 70, 105, 106

Printer Preferences: 38

VT100 Modes: 28, 31, 34, 35, 37, 103 Personal Preferences: 22, 52, 63, 70, 107 Procedure Compiler Preferences: 75, 81

Set Saved Screens Buffer Size: 7

Load Macro Keys From Disk: 27, 107, 132

Save Macro Keys To Disk: 27, 107 Load Settings From Disk: 65, 106

Save Settings To Disk: 65

# **Index Of Procedure Commands**

@ 83 ADD 87 ELAPSED 89 ADD TO MENU 119 EMPTY 89 ALERT1 83 ENABLE MENU 120 ALERT2 83 END 111 ALERT3 83 ERASE 89 **ALT 105** ERASE ALL 89 **AND 87** FASTDUMP 96 ANSWERBACK 103 FILL\$ 90 ATTACH 110 FONT 105 **BBAR 103 GBAR 106** BELL 84 GETALL 99 CLEAR 84 GETBOX \* (see note below) CLOSE 96 GETCOST 91 COMM 104 GETFILE 100 COMPILE LOUD 99 GETGLOBAL 90 COMPILE QUIET 99 GETGROUP \* (see note below) CONCAT 87 GETLINE 85 CONTAINS 88 GETPARAM 134 CONTROL1 ^ 104 GETSELECT 99 CONTROL2 ^ 104 GOSUB 112 HOST 112 CONTROL3 ^ 104 IF ERROR 112 CONVUP 88 IF NO 113 COPYINTO 88 IF NO ERROR 113 CRAWL 111 **IF YES 113** CRC 104 INSERT MENU 120 DEFINE MENU 120 INSTR 91 DELETE 99 JUMPTO 113 DELKEY 105 LF 106 DIAL 84 LEFT\$ 91 DISABLE MENU 120 LENGTH 91 DISPLAY 105 LET EQUAL 92 DIVIDE 88 LOADSET 106 DO 111 LOADVAR 92 **ECHO 103** LOCK 114

| 1070 5557 05     |                             |  |  |
|------------------|-----------------------------|--|--|
| LONG BREAK 85    | RIGHT\$ 93                  |  |  |
| LOUD 106         | RUN 117                     |  |  |
| MACRO 107        | SAVETIME 94                 |  |  |
| MBAR 107         | SAVEVAR 94                  |  |  |
| MENUDOES 121     | SCREEN 108                  |  |  |
| MENUOFF 121      | SCREENDISK 101              |  |  |
| MID\$ 92         | SCREENPRINT 103             |  |  |
| MODEM 107        | SENDA 98                    |  |  |
| MULTIPLY 92      | SENDCLIP 99                 |  |  |
| NOZAP 107        | SENDK 98                    |  |  |
| NUMTOSTRING 93   | SENDX 98                    |  |  |
| ONPANIC 115      | SENDY 98                    |  |  |
| OR 93            | SETCOST 109                 |  |  |
| PANICAFTER 115   | SETGROUP * (see note below) |  |  |
| PAUSE 116        | SETBOX * (see note below)   |  |  |
| PRINTER 108      | SHELL * (see note below)    |  |  |
| PROMPT 85        | SHORT BREAK 85              |  |  |
| PROMPT ^ 85      | SHOW@ 83                    |  |  |
| PUTFILE 100      | SLOW 109                    |  |  |
| PUTPARAM 134     | STRINGTONUM 94              |  |  |
| QUERY1 121       | STRIP 109                   |  |  |
| QUERY2 122       | SUBTRACT 95                 |  |  |
| QUERY3 122       | TEST 95                     |  |  |
| QUERY4 123       | TIMEDAȚE 95                 |  |  |
| QUERY5 123       | TTY 109                     |  |  |
| QUIET 108        | TYPE 86                     |  |  |
| QUIT 116         | USERCLOSE 101               |  |  |
| RECA 97          | USEROPENA 101               |  |  |
| RECK 97          | USEROPENI 101               |  |  |
| RECX 98          | USEROPENO 102               |  |  |
| RECY 98          | USERREAD 102                |  |  |
| REDIAL 84        | USERWRCR 102                |  |  |
| REDIAL LIMIT 108 | USERWRITE 102               |  |  |
| RENAME 100       | VBAR 110                    |  |  |
| REPLACE\$ 93     | VT100 109                   |  |  |
| RESET 108        | VT52 109                    |  |  |

RETURN 117

WAIT 118

WATCH 86 WRITE 103 XKSTRIP 110

\* Note: These commands are not documented in this manual due to their complexity. They are used in conjunction with a separate compiler to produce a customized communications interface. The compiler and documentation will be distributed through the Red Ryder Library in the FreeSoft RoundTable on GEnie. Those who do not have access to GEnie may request the compiler and documentation by sending \$10 (\$15 outside of the United States or Canada). Please mention with your payment that you wish to order the "Shell Compiler" and related materials.

| ŧ        |      |   |   |
|----------|------|---|---|
| •        |      |   |   |
| 1        |      |   |   |
|          |      |   |   |
| <b>T</b> |      |   | • |
|          |      |   |   |
| 1        |      |   |   |
| •        |      |   |   |
| 1        |      |   |   |
| <u> </u> |      |   |   |
| 1        |      |   |   |
| •        |      |   |   |
| •        |      |   |   |
| 5        |      | - |   |
|          |      |   |   |
|          |      |   |   |
|          |      |   |   |
| <b>.</b> |      |   |   |
|          |      |   |   |
|          |      |   |   |
|          |      |   |   |
| -<br>1   |      |   |   |
|          | <br> |   |   |